Thinking with Imitation: Adaptive Reinforcement-Imitation Learning for Tool-Augmented Scientific Reasoning

Fanrui Zhang[1,2,*] **Hongmin Zhan**[1,*] **Qiang Zhang**[1] **Sizhuo Zhou**[1] **Zhaoyu Wang**[3]
Kaipeng Zhang[2] **Jiawei Liu**[1,†] **Zheng-Jun Zha**[1]

[1]University of Science and Technology of China, China
[2]Shanghai Innovation Institute, China
[3]Hunan University, China

Abstract

Expert scientists rarely solve genuinely novel problems through parametric recall alone; instead, they reason by analogy to structurally related prior cases and adapt their solution procedures to new contexts. Yet current multimodal large language models (MLLMs), even when equipped with retrieval, computation, and visual tools, still struggle with this imitation-driven form of reasoning. We further observe a counterintuitive negative result: simply prepending expert problems and full solution trajectories as context does not improve performance on novel scientific tasks, and can even slightly degrade it. Motivated by this observation, we argue that scientific reasoning should shift from passive answer conditioning to an imitation-driven paradigm, in which models actively seek relevant precedents and reuse their solution structures. We instantiate this idea with MimicAgent, a tool-augmented scientific reasoning agent that retrieves structurally related exemplars, interprets their reasoning trajectories, and transfers their solution patterns to the target problem through multi-step reasoning and tool interaction. To train this behavior under sparse rewards, we introduce ARIS (Adaptive Reinforcement-Imitation Switching), which unifies on-policy reinforcement learning with online expert imitation. When all rollouts for a prompt fail, ARIS replaces degenerate reinforcement updates with expert-generated demonstrations, yielding an implicit imitation-to-reinforcement curriculum without requiring a cold-start stage or fixed-ratio offline mixing. To support training and evaluation, we further introduce SciExplore-Bench, a bilingual multimodal benchmark for open-ended experimental scientific reasoning, paired with a manually curated repository of analogical exemplars. The substantial gains and impressive generalization across multiple benchmarks suggest that imitation serves both as a global paradigm for exemplar-guided reasoning and as a local mechanism for stabilizing reinforcement learning. The code is available at https://github.com/zfr00/MimicAgent.

1 Introduction

Expert scientists rarely solve genuinely novel problems through parametric recall alone. Instead, they reason by analogy with structurally related prior cases, identifying which solution schema applies, mapping its variables to the new situation, and transferring its procedural structure. Whether multimodal large language models (MLLMs) can perform such exemplar-guided, tool-augmented

[*]Equal contribution.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

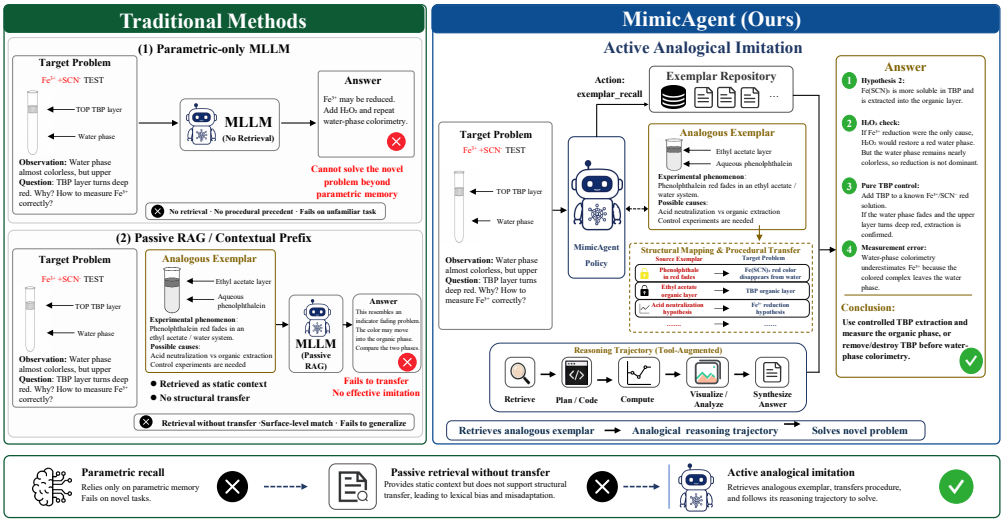


Figure 1: Traditional MLLMs either rely on parametric memory or treat retrieved exemplars as static context, which fails to support effective analogical transfer and imitation. MimicAgent instead retrieves a structurally analogous precedent and transfers its procedural structure to the target task through tool-augmented reasoning. In this example, the MimicAgent maps the source exemplar's phase-partition phenomenon, competing-cause analysis, control-experiment design, and measurement-correction strategy onto a new Fe^{3+}–SCN^{-}/TBP experimental setting, yielding a grounded solution rather than surface-level retrieval.

reasoning on problems beyond their training coverage remains unclear. The question is especially acute for experimental scientific inquiry, where solving a problem requires identifying variables and controls, interpreting visual evidence, inferring latent mechanisms from partial observations, and justifying conclusions in natural language, all tasks where expert behavior is dominated by analogical transfer rather than memorization. Despite progress on multimodal evaluation, existing scientific benchmarks capture only a limited slice of this reasoning. Datasets such as MathVista [28] and MMMU [53] broadened multimodal coverage but remain centered on closed-form question answering, factual recall, or fixed-format reasoning, where success can be obtained without any cross-problem transfer. Authentic experimental inquiry, in contrast, requires designing experiments, identifying variables and controls, interpreting multimodal evidence, and inferring hidden mechanisms from partial observations. These tasks are inherently open-ended, making analogical transfer essential, yet they remain underrepresented in current benchmarks.

To address this gap, we introduce SciExplore-Bench, a bilingual multimodal benchmark for open-ended experimental scientific reasoning. Each problem is rewritten into an open-form scientific inquiry task that requires full analysis rather than answer selection, and is paired with a curated repository of analogical exemplars aligned by shared scientific principles, experimental structures, and transferable reasoning patterns rather than superficial lexical similarity. To minimize contamination and probe post-training generalization, the evaluation split is drawn exclusively from 2026 sources beyond the likely training horizon of current frontier models. SciExplore-Bench thus jointly emphasizes open-ended experimental reasoning, multimodal evidence interpretation, bilingual evaluation, and exemplar-grounded generalization. To study this task setting, we build a unified tool-augmented agent framework that equips models with web search, localized visual inspection, structured knowledge graph querying, code execution, and exemplar retrieval. Our experiments on SciExplore-Bench reveal two key findings. First, equipping frontier MLLMs with retrieval, computation, and visual tools yields little improvement over tool-free inference, indicating that access to tools does not automatically translate into effective tool use. Second, even directly prepending manually curated analogical exemplars as static context fails to produce consistent gains. Together, these results suggest that the true bottleneck is neither the lack of relevant precedents nor the lack of tools, but the absence of a learned behavior for deciding when retrieval is useful, which structural elements of a precedent should be reused, and how its solution schema should be mapped to the target problem.

We therefore argue for an imitation-driven paradigm, in which imitation plays two complementary roles within the same training loop. As shown in Figure 1, the key distinction is not whether retrieval is present, but whether retrieved exemplars can be used as transferable procedural precedents. Parametric-only MLLMs fail when the target problem lies beyond their memorized knowledge, whereas passive RAG or contextual-prefix baselines still treat exemplars as static evidence and therefore fail to support effective analogical transfer and imitation. *Globally*, we propose MimicAgent, a tool-augmented agent in which exemplar retrieval is modeled as a policy action rather than a static context prefix. Beyond `exemplar_recall` for exemplar retrieval, MimicAgent has access to web search, structured knowledge lookup, code execution, and localized image inspection, and decides at each step which tool to invoke. The key novelty over conventional retrieval-augmented generation [24] is that we reward not only the retrieval event itself but also the procedural use that follows it: an outcome-conditioned reward R_{analogy} uses a separate reward model to credit structural identification, variable/role mapping, and procedural transfer in the post-retrieval reasoning trace, so the agent is incentivized only when retrieved precedents are actually used as transferable solution schemas. Retrieval thus becomes a learned mechanism for acquiring transferable reasoning behavior rather than a passive supplier of evidence. *Locally*, we propose ARIS (Adaptive Reinforcement-Imitation Switching), which adaptively replaces the on-policy update with an online imitation update precisely on the prompts where reinforcement learning would otherwise stall. Concretely, when every rollout in a GRPO [41] group is incorrect (a regime that dominates early training on novel scientific problems), the group-relative advantages collapse to zero, and the gradient becomes uninformative. ARIS detects these zero-answer groups and substitutes the degenerate update with a token-masked imitation loss on a teacher-generated trajectory drawn from the same tool schema. As the student improves, the fraction of zero-answer groups falls and the imitation branch decays automatically, yielding an implicit imitation-to-RL curriculum that requires neither cold-start SFT nor a fixed offline-online mixture. As we show empirically, this adaptive switching substantially outperforms both static SFT and conventional GRPO, because it concentrates expert supervision exactly on the failure modes the current policy actually exhibits.

Experimental results show that MimicAgent improves over its backbone by more than 50% relative accuracy on the scientific reasoning benchmarks SciExplore-Bench and SceMQA, surpassing strong frontier closed-source MLLMs such as GPT-5.2 and Gemini-3-Flash. It further exhibits striking generalization on the analogical reasoning benchmarks MM-IQ and VisualPuzzles. These results suggest that imitation serves both as a global paradigm for precedent-guided reasoning and as a local mechanism for stabilizing reinforcement learning. We summarize our contributions as follows. (1) We introduce SciExplore-Bench, a bilingual multimodal benchmark of open-ended experimental scientific reasoning paired with a curated analogical exemplar repository aligned by scientific structure rather than lexical overlap. (2) We propose MimicAgent, which makes exemplar-guided procedural imitation a learned inference behavior by modeling retrieval as a policy action and rewarding the use of retrieved precedents rather than retrieval itself. (3) We develop ARIS, an adaptive reinforcement-imitation algorithm that replaces collapsed GRPO updates with online expert imitation, yielding an implicit imitation-to-RL curriculum without cold-start trajectories or fixed offline mixtures.

2 Related Works

Multimodal Scientific Reasoning Benchmarks. Multimodal scientific reasoning has been studied through a rapidly growing suite of benchmarks. Early efforts such as ScienceQA [29] and MathVista [28] pioneered visual reasoning in scientific contexts, while broader academic evaluation suites [53, 21, 58, 10] target multidiscipline understanding and reasoning. Subsequent benchmarks further increased the level of difficulty, covering graduate-level science and reasoning [38, 49, 16], Olympiad-level multimodal mathematics [18, 55], free-response college-entrance science [27], and tool-augmented agentic evaluation [30, 42]. Overall, these benchmarks have progressively expanded the scope of multimodal scientific reasoning, from visual question answering and academic testing to more challenging open-ended and tool-augmented settings.

Reinforcement and Imitation Learning for LLM Reasoning. RLHF based on PPO [40, 34] pioneered preference-driven training for large language models, with DPO [37] and RLOO [1] providing simpler and more efficient alternatives. For reasoning tasks with verifiable rewards, GRPO [41] replaces the value baseline with a group-relative estimator and underlies recent advances such as DeepSeek-R1 [17], DAPO [51], and VinePPO [22]. A persistent challenge of these group-relative methods is that, on sufficiently difficult prompts, all rollouts may fail simultaneously, causing the

group advantage to collapse and the gradient to vanish. This issue has become particularly visible on recent abstract-reasoning benchmarks such as MM-IQ [9], VisualPuzzles [45], and Induction-Bench [20], which reveal persistent gaps in perception-reasoning coupling and structural induction. Imitation learning provides a complementary line of work, from DAgger [39] and expert iteration [2] to advantage-weighted regression [35], with recent LLM-oriented variants such as ReST$^{\mathrm{EM}}$ [43] and Self-Rewarding Language Models [52].

3 SciExplore-Bench Dataset Construction

The construction pipeline of SciExplore-Bench (Figure 2) involves four stages: bilingual data collection, experiment-oriented filtering, open-ended rewriting, and analogical exemplar curation.

Data Collection. We collect a bilingual corpus of multimodal science problems from Chinese and English educational sources accessible via web search, with an approximately $1{:}1$ balance between the two languages. The corpus spans physics, chemistry, and biology, and is drawn from advanced high-school ($\sim 70\%$) and undergraduate-level ($\sim 30\%$) materials. Approximately 70% of the problems contain images, including experimental apparatus diagrams, data plots, microscopic illustrations, and schematic figures. We additionally retain auxiliary metadata, such as knowledge-point tags and detailed solution analyses.

Experiment-Oriented Filtering. To isolate problems that genuinely involve experimental scientific inquiry, we score each candidate with `GPT-5.2` along five dimensions: (i) presence of experimental design, procedures, or operations; (ii) the requirement to propose plans, predict outcomes, or analyze observations; (iii) explicit experimental materials, conditions, or variable-control structures; (iv) the need to interpret data and draw scientific conclusions; and (v) explicit reasoning over experimental processes. A problem is retained only if it satisfies at least one of these criteria.

Open-Ended Rewriting. Each retained problem is rewritten into an open-ended inquiry task: closed-form cues such as multiple-choice options and fill-in blanks are removed; information that should be inferred or acquired is withheld; and the underlying scientific principles are folded back into the experimental scenario while preserving the original scientific question. Rewriting is paired with a self-verification loop that checks answer completeness, absence of answer leakage, and consistency between the rewritten task and its reference solution; only verified samples enter the dataset.

Analogical Exemplar Repository. A defining feature of SciExplore-Bench is the analogical exemplar repository attached to each target problem. We exploit the pedagogical structure of authoritative science textbooks: high-quality materials in both Chinese and English organize their exercises into instructional units in which a target problem is accompanied by companion exercises deliberately selected by domain experts, namely textbook editors and senior teachers, to reinforce the same scientific principle, experimental design, and solution schema. Treating these expert-organized companion exercises as procedural exemplars yields a repository whose alignment is *curatorial* rather than lexical or embedding-based: exemplars are structurally compatible at the level of scientific content, not surface form. This property is preserved through the rewriting stage, since rewriting alters form rather than the underlying schema. To minimize contamination from the likely training horizon of frontier models, the test split is drawn *exclusively* from sources published in 2026. This temporal separation yields 480 evaluation problems beyond plausible parametric memorization, while still providing $6{,}000$ rewritten training problems under the same task formulation for supervised and reinforcement-learning experiments. Additional benchmark analysis is provided in Section C.

4 Method

4.1 Overview of MimicAgent

As shown in Figure 2, MimicAgent realizes the imitation-driven paradigm via three components: a unified tool suite that defines its action space, an outcome-conditioned multi-dimensional reward that makes procedural imitation a learnable training target, and **ARIS** (**A**daptive **R**einforcement-**I**mitation **S**witching), which replaces degenerate on-policy updates with online expert imitation when group-relative learning signals collapse. We detail each component below.

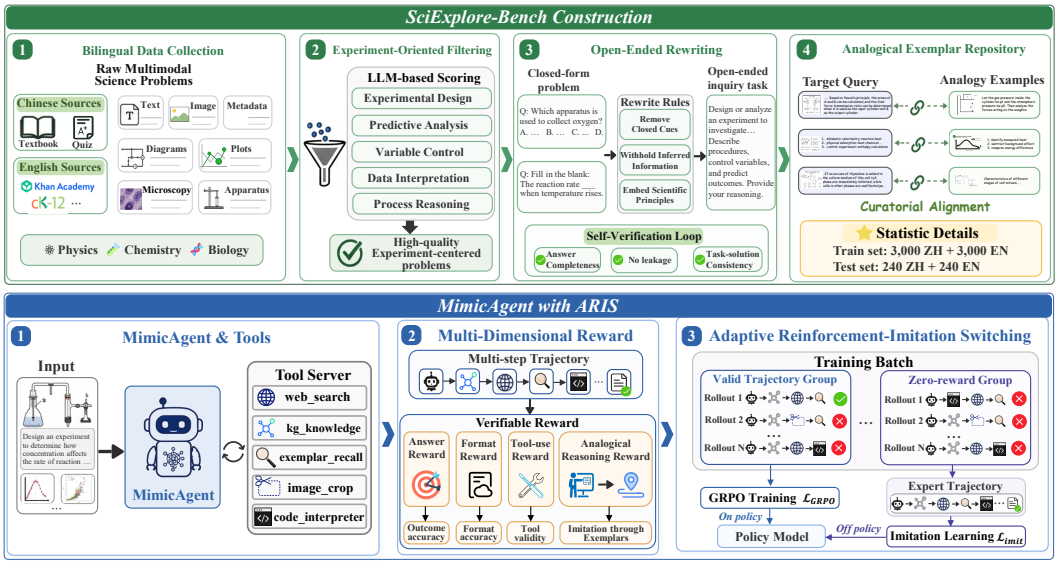


Figure 2: The overall architecture is illustrated, with the upper part showing the SciExplore-Bench construction process and the lower part presenting the training pipeline of MimicAgent.

4.2 Tool Suite for Scientific Exploratory Reasoning

To support imitation-driven scientific reasoning, we equip MimicAgent with a unified tool suite

$$\mathcal{T} = \{t_1, \ldots, t_5\}, \tag{1}$$

where each tool maps structured arguments to observations through a shared JSON interface. The suite covers five complementary functions required for open-ended scientific exploration: exemplar retrieval, open-domain evidence acquisition, structured knowledge lookup, verifiable computation, and localized visual inspection.

Among these tools, `exemplar_recall` is the most distinctive component, as it enables *exemplar-guided analogical reasoning*. Unlike conventional RAG, which treats retrieval as similarity-based evidence augmentation, our `exemplar_recall` performs an *indexed lookup* into the curated analogical exemplar repository introduced in Section 3: given a target problem, the call deterministically returns $K{=}3$ pre-aligned companion exercises together with their reference solution traces. Because alignment is curatorial rather than embedding-based, retrieval quality is not bounded by query-document similarity, and learning therefore concentrates entirely on *when* to invoke retrieval and *how* to use the returned exemplars rather than on query formulation. `exemplar_recall` thus becomes a learned mechanism for procedural imitation: the agent must decide when consulting prior cases is beneficial and how to transfer their solution structure into valid reasoning steps for the new problem. The remaining tools provide complementary support. `web_search` supplies open-domain scientific evidence when external context is needed. `kg_knowledge` performs multi-hop retrieval over the ATLAS-Wiki knowledge graph [6] and returns structured evidence-conclusion pairs, with `gpt-4.1-mini` used as a bilingual translator and filter. `code_interpreter` executes Python code in a controlled environment with standard mathematical libraries, extending the model from approximate symbolic reasoning to verifiable numerical computation. Finally, `image_crop` enables region-level visual inspection by cropping task-relevant image patches and re-injecting them as multimodal context. Together, these tools define the operational interface through which MimicAgent carries out exemplar-guided, tool-augmented scientific reasoning.

4.3 Outcome-Conditioned Multi-Dimensional Reward

To train MimicAgent as a tool-augmented scientific reasoning agent, we adopt an agentic GRPO framework, in which the policy is optimized over multi-step trajectories consisting of reasoning, tool invocation, and environment feedback. In this setting, reward design is critical: the agent must be encouraged not only to produce correct final answers, but also to retrieve useful exemplars,

invoke tools effectively, and generate verifiable intermediate computations. We therefore design an outcome-conditioned multi-dimensional reward that aligns process-level behaviors with end-task success while making imitation-driven retrieval a learnable objective.

The overall reward is

$$R_{\text{total}} = w_a R_{\text{answer}} + w_f R_{\text{format}} + w_t R_{\text{tool}} + w_r R_{\text{analogy}} \tag{2}$$

Answer reward. $R_{\text{answer}} \in \{0, 1\}$ measures final answer correctness under LLM-based judgment and serves as the anchor signal for all other reward terms.
Format reward. $R_{\text{format}} \in \{0, 1\}$ checks whether the trajectory follows the required schema, with tool calls, and predictions enclosed in `<tool_call>`, and `<answer>` tags respectively.
Tool-use reward. R_{tool} rewards correctly parsed and successfully executed tool calls by assigning fine-grained credit to valid tool usage, while invocations that repeat the same tool with identical or near-identical arguments receive no additional credit. To discourage frequent but unhelpful tool use, this reward is further conditioned on the final answer outcome: a successful call retains full credit when the trajectory ends in a correct answer ($R_{\text{answer}} = 1$), and is scaled by 0.2 when the final answer is judged incorrect. This conditioning makes tool-use credit primarily outcome-aligned, while retaining a weak residual signal for potentially useful tool behaviors in failed trajectories. A detailed description of the implementation is provided in Section A.
Analogical reasoning reward. To explicitly encourage *imitation through exemplars*, we reward not the retrieval event itself, but the model's ability to *use* a retrieved exemplar as a transferable procedural precedent. Concretely, once the agent invokes `exemplar_recall`, we score the subsequent reasoning trace with a closed-source reward model that evaluates whether the model has successfully transferred the exemplar's solution structure to the target problem.

The reward model checks for three types of evidence of procedural imitation: (i) *structural identification*, namely recognizing a relevant solution schema in the retrieved exemplar; (ii) *role mapping*, namely aligning the variables, entities, or experimental roles in the exemplar with those in the target problem; and (iii) *procedural transfer*, namely applying the exemplar's solution procedure to produce valid downstream reasoning steps. Since any one of these behaviors already indicates partial imitation, the reward is given whenever at least one of them is present, and increases with stronger evidence of transfer. Let s_{id}, s_{map}, and s_{trans} denote the normalized scores assigned by the reward model to these three dimensions. We define the base analogical reward as

$$R_{\text{analogy}} = \max(s_{\text{id}},\, s_{\text{map}},\, s_{\text{trans}})\,. \tag{3}$$

R_{analogy} rewards not retrieval itself, but the successful imitation of transferable solution structure. It is further subject to the same outcome-conditioning rule as R_{tool}.

4.4 ARIS: Adaptive Reinforcement-Imitation Switching

While the outcome-conditioned reward makes exemplar retrieval and tool use learnable, optimizing such behaviors with standard on-policy reinforcement learning remains highly unstable. In our setting, the base model often fails on genuinely novel scientific problems, and sparse rewards provide almost no usable signal when all sampled rollouts are incorrect. This issue is amplified in tool-augmented environments, where reasoning errors and tool failures accumulate over multiple steps. As a result, naive GRPO-style training frequently produces *zero-answer groups*, wasting optimization exactly on the prompts where additional supervision is most needed. Rather than discarding these groups or mixing in offline demonstrations at a fixed ratio, we treat them as opportunities for adaptive expert intervention. We next describe ARIS in detail.

GRPO estimates within-group relative advantages and optimizes the clipped surrogate

$$\begin{aligned} \hat{A}_i &= \frac{r_i - \mu(\mathbf{r})}{\sigma(\mathbf{r}) + \epsilon}, \\ \mathcal{L}_{\text{GRPO}}(q) &= -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\big(\rho_{i,t}\hat{A}_i,\ \text{clip}(\rho_{i,t}, 1-\epsilon_l, 1+\epsilon_h)\hat{A}_i\big), \end{aligned} \tag{4}$$

where $\rho_{i,t}$ is the importance ratio. Following DAPO [51], we use asymmetric clipping $(\epsilon_l, \epsilon_h) = (0.20, 0.28)$, and set the KL coefficient to zero. Let $c_i \in \{0, 1\}$ denote the answer correctness of

rollout o_i under LLM-based judgment, and define

$$I_{\text{zero}}(q) = \mathbf{1}\left[\max_i c_i = 0\right]. \tag{5}$$

When all rollouts in a group are incorrect, $I_{\text{zero}}(q) = 1$, the answer-correctness signal collapses, and $\hat{A}_i \approx 0$, yielding almost no useful gradient.

To address this, we introduce ARIS, which unifies reinforcement learning and expert imitation under a shared weighted log-likelihood view in the spirit of advantage-weighted regression:

$$\mathcal{L}_{\text{AWR}}(\theta) = -\mathbb{E}_{(q,\tau)\sim\mathcal{D}}\left[w(q,\tau)\cdot\log\pi_\theta(\tau \mid q)\right]. \tag{6}$$

Under this formulation, GRPO corresponds to on-policy student trajectories weighted by normalized group advantages, whereas imitation corresponds to off-policy expert trajectories with uniform weights. Concretely, the imitation objective is defined as

$$\mathcal{L}_{\text{imit}}(q) = -\frac{1}{|y^*|}\sum_{t=1}^{|y^*|}\log\pi_\theta(y_t^* \mid q, y_{<t}^*), \tag{7}$$

where y^* is the expert trajectory generated by the teacher policy for prompt q. In practice, $\mathcal{L}_{\text{imit}}$ is computed only on model-generated tokens in the expert trajectory, while tool-response tokens are masked out. Based on this unified view, ARIS performs one weighted policy update per prompt and switches between the two regimes according to the validity of the on-policy signal:

$$\mathcal{L}_{\text{ARIS}}(q) = \big(1 - I_{\text{zero}}(q)\big)\cdot\mathcal{L}_{\text{GRPO}}(q) + I_{\text{zero}}(q)\cdot\lambda\cdot\mathcal{L}_{\text{imit}}(q), \tag{8}$$

where $\lambda > 0$ is the scaling factor for the imitation branch. If the sampled group contains at least one correct rollout, ARIS reduces to standard GRPO; if all rollouts fail, the GRPO update is discarded and replaced by imitation on an expert trajectory. The expert trajectory y^* is generated online by an independent teacher policy $y^* \sim \pi_T(\cdot \mid q; \mathcal{T}, \mathcal{F})$, where the tool set $\mathcal{T}$ and formatting constraints $\mathcal{F}$ are identical to those of the student. We instantiate π_T with a stronger closed-source model, ensuring that each demonstrated action remains valid under the student's tool schema. For efficiency, we use prompt engineering together with partial ground-truth injection to elicit high-quality teacher trajectories; a detailed analysis of the prompting strategy is provided in Section E.1.

Although the GRPO and imitation branches are mutually exclusive at the prompt level, the expected batch loss of ARIS remains a weighted mixture:

$$\mathbb{E}_q[\mathcal{L}_{\text{ARIS}}(q)] = \big(1 - p_{\text{zero}}^{(k)}\big)\cdot\mathbb{E}[\mathcal{L}_{\text{GRPO}}] + p_{\text{zero}}^{(k)}\cdot\lambda\cdot\mathbb{E}[\mathcal{L}_{\text{imit}}], \tag{9}$$

where $p_{\text{zero}}^{(k)}$ is the proportion of zero-answer groups at training step k. As training progresses and the student improves, $p_{\text{zero}}^{(k)}$ naturally decreases, causing the imitation branch to vanish automatically. ARIS therefore yields an implicit imitation-to-RL curriculum, stabilizing early learning while preserving purely on-policy optimization in later stages.

5 Experiments

5.1 Experimental Settings

Dataset. For evaluation, we use the bilingual test split of SciExplore-Bench (CN and EN) and 150 free-response questions from the physics, chemistry, and biology subsets of SceMQA [27] as an external benchmark for scientific experimental reasoning.

Baselines. We compare MimicAgent against four groups of competitive baselines, all evaluated under the tool-augmented setting unless explicitly marked *w/o tools*: (i) *open-source MLLMs*—InternVL2.5-8B/26B [12], Qwen2.5-VL-7B/32B [8], and Qwen3-VL-8B/30B [7]; (ii) *closed-source MLLMs*—GPT-5-mini, GPT-5.2, Claude-Haiku-4.5, Gemini-2.5-Flash, Claude-Sonnet-4.5, and Gemini-3-Flash; (iii) *tool-ablated variants* of representative open- and closed-source models, where tool access is disabled to isolate the contribution of tool use (*w/o tools*), while exemplars are supplied as a fixed context prefix (*prefix*). (iv) *post-training baselines* sharing the same backbone and training data as MimicAgent: SFT, GRPO [41], DAPO [51], GSPO [57], and SFT+GRPO.

Implementation Details. All experiments are implemented in PyTorch on 8 H100 GPUs, using

Table 1: Main results on SciExplore-Bench (CN, EN) and SceMQA. We report accuracy (%) on the physics (Phy.), chemistry (Chem.), and biology (Bio.) subsets of each benchmark.

Model	SciExplore-Bench CN			SciExplore-Bench EN			SceMQA			Overall
	Phy.	Chem.	Bio.	Phy.	Chem.	Bio.	Phy.	Chem.	Bio.	Acc.
InternVL2.5-8B [12]	30.00	28.75	36.25	27.50	22.50	32.50	18.00	28.00	48.00	30.00
InternVL2.5-26B [12]	42.50	33.75	35.00	36.25	43.75	41.25	20.00	30.00	42.00	36.83
Qwen2.5-VL-7B [8]	30.00	28.75	37.50	26.25	30.00	38.75	26.00	34.00	50.00	33.02
Qwen2.5-VL-32B [8]	37.50	38.75	41.25	43.75	52.50	45.00	12.00	8.00	52.00	38.57
Qwen3-VL-8B [7]	18.75	25.00	20.00	37.50	56.25	47.50	36.00	42.00	22.00	33.97
Qwen3-VL-30B [7]	58.75	62.50	65.00	45.00	50.00	47.50	56.00	70.00	62.00	56.67
GPT-5-mini [32]	72.50	70.00	78.75	70.00	67.50	76.25	72.00	80.00	70.00	72.86
GPT-5.2 [32]	77.50	68.75	82.50	71.25	66.25	80.00	72.00	84.00	68.00	74.44
Claude-Haiku-4.5 [4]	45.00	51.25	72.50	55.00	67.50	63.75	68.00	74.00	56.00	60.79
Claude-Sonnet-4.5 [5]	57.50	52.50	66.25	53.75	53.75	65.00	72.00	78.00	64.00	61.27
Gemini-2.5-Flash [14]	71.25	68.75	76.25	76.25	70.00	75.00	72.00	78.00	70.00	73.02
Gemini-3-Flash [15]	80.00	68.75	82.50	80.00	62.50	81.25	72.00	78.00	74.00	75.56
Claude-Haiku-4.5 (w/o tools)	68.75	60.00	58.75	57.50	58.75	68.75	64.00	72.00	58.00	62.70
GPT-5-mini (w/o tools)	66.25	53.75	76.25	50.00	62.50	73.75	72.00	80.00	60.00	65.40
Claude-Haiku-4.5 (prefix)	57.50	68.75	75.00	60.00	57.50	71.25	68.00	68.00	60.00	65.08
Qwen3-VL-30B (prefix)	51.25	55.00	61.25	47.50	38.75	41.25	66.00	68.00	58.00	52.70
Qwen2.5-VL-32B (prefix)	36.25	42.50	41.25	45.00	42.50	55.00	12.00	10.00	48.00	38.89
SFT [34]	48.75	58.75	70.00	56.25	62.50	67.50	48.00	52.00	60.00	58.89
GRPO [41]	60.00	55.00	73.75	56.25	65.00	71.25	76.00	72.00	70.00	65.71
DAPO [51]	52.50	55.00	76.25	62.50	63.75	67.50	74.00	74.00	68.00	65.08
GSPO [57]	52.50	56.25	71.25	56.25	60.00	77.50	70.00	78.00	70.00	64.76
SFT+GRPO [41]	76.25	63.75	80.00	65.00	75.00	75.00	78.00	72.00	72.00	72.86
MimicAgent	**88.75**	**88.75**	**93.75**	**82.50**	**82.50**	**88.75**	**80.00**	**86.00**	**78.00**	**86.03**

Qwen3-VL-8B-Instruct as the base model. We sample $G{=}8$ rollouts per prompt at temperature 0.8, with up to 6 rounds of tool interaction, and set the imitation scaling to $\lambda{=}10$ and the four reward weights to $w_a{=}0.40$, $w_f{=}0.15$, $w_t{=}0.25$, $w_r{=}0.20$ (summing to 1, so $R_{\text{total}}{\in}[0,1]$). The teacher policy π_T is instantiated with `gpt-5.2`, while the reward model for R_{analogy} is instantiated with `gemini-3-flash`. The full training schedule is reported in Section A.

5.2 Main Results

Comparison to State-of-the-Art Models. Table 1 shows that MimicAgent achieves the best overall performance on SciExplore-Bench CN, SciExplore-Bench EN, and SceMQA. Closed-source models consistently outperform open-source ones, with Gemini-3-Flash providing the strongest proprietary baseline. A key observation comes from the experiments on tool use and exemplar usage. The *w/o tools* variants perform similarly to their tool-enabled counterparts, indicating that simply providing current models with retrieval, knowledge, computation, and visual tools does not automatically translate into effective tool use. Meanwhile, the *exemplar prefix* variants, which prepend manually curated analogical exemplars as static context, also fail to produce consistent gains. We further evaluate different training strategies on the same training set to isolate the effect of ARIS. SFT uses teacher-generated trajectories for all queries, GRPO, DAPO, and GSPO directly optimize the policy with reinforcement learning, and SFT+GRPO combines supervised cold start with subsequent policy optimization. Although these methods improve over the base open-source model, they remain substantially below MimicAgent, indicating that neither static demonstrations nor conventional RL is sufficient to internalize exemplar-guided imitation as a robust reasoning skill.

Generalization. We further evaluate MimicAgent on two non-scientific visual reasoning benchmarks, MM-IQ [9] and VisualPuzzles [45], to test whether the learned behavior transfers beyond the scientific domain. Both benchmarks explicitly minimize dependence on domain knowledge: MM-IQ contains 2,710 Raven-style abstract visual reasoning problems, and VisualPuzzles contains 1,168 problems spanning algorithmic, analogical, deductive, inductive, and spatial reasoning. As shown in Table 2, MimicAgent substantially outperforms its base model Qwen3-VL-8B-Instruct (+23 on MM-IQ, +23 on VisualPuzzles) and compares favorably with published closed-source results, including o3, Claude-3.7-Sonnet, Gemini-2.5-pro, and GPT-4o. These results indicate that ARIS does not merely specialize the agent to scientific tasks, but broadly improves its visual analogical reasoning.

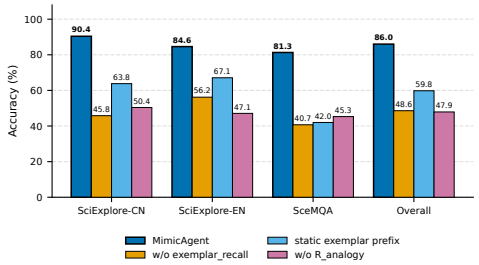
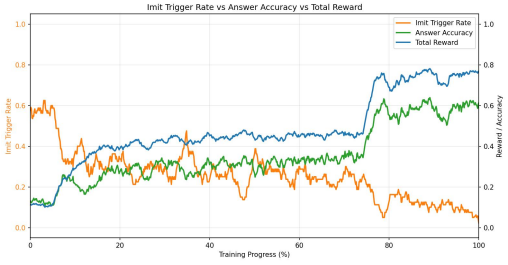


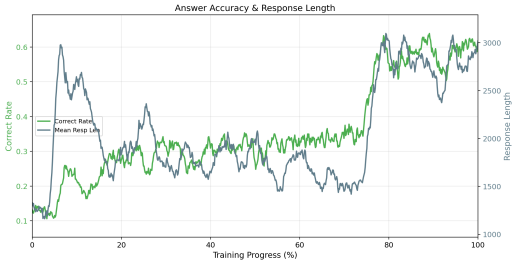
Figure 3: Ablation study of MimicAgent. Accuracy (%) averaged over the three subjects (physics/chemistry/biology).

Model	MM-IQ	VisualPuzzles
Qwen3-VL-8B [7]	26.75	36.99
o3 [33]	33.17	54.00
Claude-3.7-Sonnet [3]	31.55	48.20
Gemini-2.5-pro [14]	31.23	49.50
GPT-4o [31]	26.87	41.30
QvQ-72B-Preview [47]	26.94	37.80
MimicAgent	**50.11**	**59.93**

Table 2: Generalization on out-of-domain visual reasoning benchmarks. For each baseline we cite the published numbers reported on the corresponding benchmark.



(a) Imit-branch trigger rate (orange, left axis), answer accuracy (green, right axis), and total reward R_{total} (blue, right axis) versus training progress.



(b) Correct rate (green, left axis) and mean response length (dark blue, right axis) versus training progress.

Figure 4: zero-answer dynamics during ARIS training. (a) The imitation-branch trigger rate drops from $\sim 50\%$ to $\sim 5\%$, while answer accuracy and total reward rise, with a clear jump near 75% progress. (b) Correct rate and mean response length increase together.

5.3 Further Analysis

Ablation Study. We further conduct ablations to isolate which components of MimicAgent are responsible for the observed gains. Specifically, we examine three aspects of the design: (i) *w/o* `exemplar_recall`, where the exemplar retrieval module is removed; (ii) *static exemplar prefix*, where retrieved exemplars are provided as a fixed context prefix rather than being actively retrieved as a policy action, and the retrieval tool is disabled; and (iii) *w/o* R_{analogy}, where the analogical reasoning reward is removed from training. As shown in Figure 3, removing either the `exemplar_recall` tool or the analogical reasoning reward causes comparably severe degradation, while passively exposing exemplars as a static context prefix recovers only part of the gap. This pattern indicates that MimicAgent's gain does not come from tool availability alone, but from learning to retrieve transferable procedural precedents, use them actively rather than passively, and internalize that behavior through analogy-aware reward shaping.

Zero-Answer Ratio Dynamics. We track the evolution of the imitation-branch trigger rate (i.e., the zero-answer group ratio $p_{\text{zero}}^{(k)}$), the answer-judge accuracy, and the total reward R_{total} throughout training to verify the implicit imitation-to-RL curriculum induced by ARIS. All curves are plotted against normalized training progress (0–100%) for comparability across runs. As shown in Figure 4(a), the imitation-branch trigger rate decreases monotonically from about 50% at the beginning of training to roughly 5% at the end, while both the answer-judge accuracy and R_{total} increase steadily, with a marked jump near 75% training progress. This clear inverse coupling matches the prediction of the ARIS objective: as the student improves, the expected weight on the imitation branch decays automatically, and training naturally transitions toward purely on-policy optimization. Figure 4(b) further shows that the correct rate rises from about 15% to $\sim 60\%$, while the mean response length increases from roughly $1{,}500$ to $\sim 3{,}000$ tokens. This suggests that ARIS does not shorten reasoning for faster reward, but instead supports longer and more effective multi-step reasoning. Over the full run, the cumulative imitation-trigger rate is approximately 32%, corresponding to only 0.32 teacher-

trajectory generations per prompt on average, substantially lower than fixed-ratio offline-online mixtures. Additional experimental results are provided in Section D.

6 Conclusion

We presented an imitation-driven framework for tool-augmented scientific reasoning at two complementary scales. Globally, MimicAgent models exemplar retrieval as a policy action to encourage retrieval only when it leads to genuine schema transfer. Locally, ARIS replaces collapsed GRPO updates on zero-answer groups with token-masked imitation from teacher trajectories, yielding an implicit imitation-to-RL curriculum without cold-start or fixed offline-online mixing. We further introduced SciExplore-Bench, a bilingual benchmark of 480 open-ended experimental problems paired with a curatorial repository of analogical exemplars. An 8B MimicAgent improves over its base by more than 50% across several benchmarks, surpasses frontier closed-source MLLMs, and exhibits strong generalization beyond the scientific domain. Taken together, these results suggest that imitation is a useful principle for both exemplar-guided reasoning and stable policy optimization.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62225207, 62476260 and 62436008, the Fundamental Research Funds for the Central Universities under Grant WK2100000057.

References

[1] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, 2024.

[2] Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017.

[3] Anthropic. Claude 3.7 sonnet. `https://www.anthropic.com/news/claude-3-7-sonnet`, 2025.

[4] Anthropic. Claude haiku 4.5. `https://www.anthropic.com/news/claude-haiku-4-5`, 2025.

[5] Anthropic. Claude sonnet 4.5. `https://www.anthropic.com/news/claude-sonnet-4-5`, 2025.

[6] Jiaxin Bai, Wei Fan, Qi Hu, Qing Zong, Chunyang Li, Hong Ting Tsang, Hongyu Luo, Yauwai Yim, Haoyu Huang, Xiao Zhou, et al. Autoschemakg: Autonomous knowledge graph construction through dynamic schema induction from web-scale corpora. *arXiv preprint arXiv:2505.23628*, 2025.

[7] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[8] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[9] Huanqia Cai, Yijun Yang, and Winston Hu. Mm-iq: Benchmarking human-like abstraction and reasoning in multimodal models. *arXiv preprint arXiv:2502.00698*, 2025.

[10] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024.

[11] Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7901, 2023.

[12] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[13] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2024.

[14] Google DeepMind. Gemini 2.5: Our most intelligent ai model. `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/`, 2025.

[15] Google DeepMind. Gemini 3. `https://blog.google/products/gemini/gemini-3/`, 2025.

[16] Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

[17] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[18] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.

[19] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[20] Wenyue Hua, Tyler Wong, Fei Sun, Liangming Pan, Adam Jardine, and William Yang Wang. Inductionbench: Llms fail in the simplest complexity class. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26526–26546, 2025.

[21] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in neural information processing systems*, 36:62991–63010, 2023.

[22] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*, 2024.

[23] Jon M Laurent, Joseph D Janizek, Michael Ruzo, Michaela M Hinks, Michael J Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D White, and Samuel G Rodriques. Lab-bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

[24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[25] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, 2024.

[26] Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

[27] Zhenwen Liang, Kehan Guo, Gang Liu, Taicheng Guo, Yujun Zhou, Tianyu Yang, Jiajun Jiao, Renjie Pi, Jipeng Zhang, and Xiangliang Zhang. Scemqa: A scientific college entrance level multimodal question answering benchmark. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 109–119, 2024.

[28] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[29] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

[30] Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, and Aixin Sun. Sciagent: Tool-augmented language models for scientific reasoning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 15701–15736, 2024.

[31] OpenAI. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`, 2024.

[32] OpenAI. Gpt-5. `https://openai.com/index/introducing-gpt-5/`, 2025.

[33] OpenAI. Introducing openai o3 and o4-mini. `https://openai.com/index/introducing-o3-and-o4-mini/`, 2025.

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[35] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[36] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[37] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[38] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[39] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[41] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[42] Yujiong Shen, Yajie Yang, Zhiheng Xi, Binze Hu, Huayu Sha, Jiazheng Zhang, Qiyuan Peng, Junlin Shang, Jixuan Huang, Yutao Fan, et al. Sciagentgym: Benchmarking multi-step scientific tool-use in llm agents. *arXiv preprint arXiv:2602.12984*, 2026.

[43] Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

[44] Shreyas Singh, Kunal Singh, and Pradeep Moturi. Fathom-deepresearch: Unlocking long horizon information retrieval and synthesis for slms. *arXiv preprint arXiv:2509.24107*, 2025.

[45] Yueqi Song, Tianyue Ou, Yibo Kong, Zecheng Li, Graham Neubig, and Xiang Yue. Visualpuzzles: Decoupling multimodal reasoning evaluation from domain knowledge. *arXiv preprint arXiv:2504.10342*, 2025.

[46] Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. Scieval: A multi-level large language model evaluation benchmark for scientific research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19053–19061, 2024.

[47] Qwen Team. Qvq-72b-preview: An open-weight visual reasoning model. `https://qwenlm.github.io/blog/qvq-72b-preview/`, 2024.

[48] Minyang Tian, Luyu Gao, Shizhuo D Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, et al. Scicode: A research coding benchmark curated by scientists. *Advances in Neural Information Processing Systems*, 37:30624–30650, 2024.

[49] Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023.

[50] Xinwu Ye, Chengfan Li, Siming Chen, Wei Wei, and Robert Tang. Mmscibench: Benchmarking language models on chinese multimodal scientific problems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14621–14663, 2025.

[51] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[52] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[53] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567, 2024.

[54] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186, 2025.

[55] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024.

[56] Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*, 2023.

[57] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[58] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the association for computational linguistics: NAACL 2024*, pages 2299–2314, 2024.

A Detailed Experimental Settings

A.1 Datasets

We evaluate MimicAgent on four multimodal benchmarks that together cover three regimes: an internal benchmark (SciExplore-Bench), an external scientific question-answering benchmark for positive cross-distribution generalization (SceMQA), and two general-purpose visual reasoning benchmarks that are largely orthogonal to our training distribution (MM-IQ and VisualPuzzles). A summary is provided in Table 3.

SciExplore-Bench (Ours). SciExplore-Bench is a bilingual multimodal benchmark for open-ended experimental scientific reasoning. Each language version (Chinese and English) contains 240 problems evenly distributed over physics, chemistry, and biology (80 per subject), giving a combined evaluation pool of 480 bilingual problems. 66.2% of the questions in the Chinese-English test set include images (experimental apparatus diagrams, data plots, microscopic illustrations, and schematic figures); pedagogical difficulty labels for the Chinese split are distributed as easy / medium / hard $= 29.6\% / 53.3\% / 17.1\%$. The training pool comprises 6,000 rewritten problems (3,000 Chinese and 3,000 English) that share the rewriting pipeline of the test split but originate from disjoint sources: test problems are drawn exclusively from sources published in 2026, while training problems are drawn from pre-2026 sources. The Chinese and English splits are independently collected from Chinese and English educational sources following the same pipeline, with per-subject sizes matched between the two languages by stratified sampling. Each problem is accompanied by a curatorial repository of *four* companion exercises selected by textbook editors and senior teachers, of which the default $K=3$ are returned by `exemplar_recall`.

SceMQA [27]. SceMQA is a college-entrance-level multimodal science benchmark. We adopt the entire physics / chemistry / biology free-response subsets (50 problems per subject, 150 problems in total) as an external positive-generalization suite. This subset shares subjects with our training distribution but uses an independent answer format (short numerical / conceptual responses) and disjoint sources, so its problems cannot appear in the analogical exemplar repository, matching the zero-shot setting of the original SceMQA paper.

MM-IQ [9]. MM-IQ contains 2,710 Raven-style abstract visual reasoning problems organized into eight categories: Mathematical (936), Temporal Movement (415), 3D-Geometry (398), Logical Operation (393), 2D-Geometry (365), Spatial Relationship (118), Visual Instruction (47), and Concrete Object (38). Because MM-IQ explicitly minimizes dependence on domain knowledge, it serves as a stress test for general visual reasoning rather than scientific recall. We evaluate without tool injection.

VisualPuzzles [45]. VisualPuzzles contains 1,168 diverse visual reasoning problems spanning five categories—Spatial (286), Algorithmic (262), Analogical (211), Inductive (209), Deductive (200)—and three difficulty levels (533 / 457 / 178). Like MM-IQ, it is designed to decouple multimodal reasoning from domain knowledge and probes whether MimicAgent retains general reasoning competence far from its training distribution.

Table 3: Summary of evaluation benchmarks.

Benchmark	Split	#Questions	Subjects	Image %	Format
SciExplore-Bench	test	480	Phy / Chem / Bio	66.2%	open-ended
SceMQA (FR)	test	150	Phy / Chem / Bio	100%	open-ended
MM-IQ	test	2,710	8 abstract types	100%	4-way MCQ
VisualPuzzles	test	1,168	5 reasoning types	100%	4-way MCQ
SciExplore-Bench (training)	train	6,000	Phy / Chem / Bio	70.1%	open-ended

Table 4 reports the per-subject sizes at every stage of the SciExplore-Bench pipeline. The Chinese training pool contains 573 physics, 1,196 chemistry, and 1,231 biology problems, totalling 3,000; the Chinese test set contains 80 problems per subject for a total of 240. The English version (training 3,000 and test 240) is independently collected from English educational sources following the same pipeline, with per-subject sizes matched to the Chinese split by stratified sampling. The full benchmark therefore comprises 6,000 training problems and 480 evaluation problems.

Table 4: Per-subject sizes of the SciExplore-Bench training and test splits. The English version is independently collected from English educational sources, with per-subject sizes matched to the Chinese split by stratified sampling.

Split	Physics	Chemistry	Biology	Total
Training, CN	573	1,196	1,231	3,000
Training, EN	573	1,196	1,231	3,000
Test, CN	80	80	80	240
Test, EN	80	80	80	240
Bilingual training (CN + EN)	1,146	2,392	2,462	6,000
Bilingual evaluation (CN + EN)	160	160	160	480

A.2 Baselines

We compare MimicAgent against 22 baselines organized into four groups: general open-source MLLMs, general closed-source MLLMs, tool-ablation variants drawn from the previous two groups under two regimes (*w/o tools* and *prefix*), and training baselines that share MimicAgent's backbone and tool stack but differ only in their training objective. Below we describe each baseline individually and indicate the role it plays in our evaluation.

(1) Open-source MLLMs. This group covers six open-source MLLMs from three families. All of them are evaluated under the same five-tool stack as MimicAgent, so that performance gaps are attributable to model capability rather than tool infrastructure.

- **InternVL2.5-8B** [12]: an 8B-parameter vision–language model from the InternVL2.5 series, jointly pre-trained on large-scale interleaved image–text corpora and used here as a small open-source reference point.
- **InternVL2.5-26B** [12]: the 26B-parameter sibling of the above, used to characterize how scaling within the same family affects experiment-centered scientific reasoning.
- **Qwen2.5-VL-7B-Instruct** [8]: the instruction-tuned 7B Qwen2.5-VL release, included as a widely adopted open-source baseline at the same parameter scale as our backbone.
- **Qwen2.5-VL-32B-Instruct** [8]: the 32B instruction-tuned variant of Qwen2.5-VL, used to compare against an open-source model that is roughly $4\times$ larger than MimicAgent's backbone.
- **Qwen3-VL-8B-Instruct** [7]: the 8B instruction-tuned model from the Qwen3-VL series and the backbone of MimicAgent. Reporting it as a baseline directly quantifies the improvement contributed by ARIS-based post-training on top of the same backbone.
- **Qwen3-VL-30B** [7]: shorthand for Qwen3-VL-30B-A3B-Instruct, a 30B mixture-of-experts variant of Qwen3-VL with $\sim$ 3B activated parameters per token, included as the strongest open-source MLLM in our evaluation pool.

(2) Closed-source MLLMs. This group contains six closed-source MLLMs, all evaluated at once during 2026 through the official API to avoid incomparability caused by vendor side model drift.

- **GPT-5-mini** [32]: the lightweight tier of OpenAI's GPT-5 family, included as a strong but cost-efficient closed-source reference.
- **GPT-5.2** [32]: the full GPT-5.2 model used through its public API; included as a widely recognized and strong general-purpose closed-source baseline.
- **Claude-Haiku-4.5** [4]: Anthropic's small-tier Claude-4.5 model, accessed through the Claude API and included as a representative compact closed-source MLLM.
- **Claude-Sonnet-4.5** [5]: Anthropic's mid-tier Sonnet model from the Claude-4.5 family, accessed through the Claude API; reported alongside Claude-Haiku-4.5 to characterise within-vendor capacity scaling on experiment-centred scientific reasoning.
- **Gemini-2.5-Flash** [14]: Google's fast tier Gemini-2.5 model, used as a closed-source baseline tuned for low-latency reasoning.

- **Gemini-3-Flash** [15]: the corresponding Gemini-3 fast tier, included as a prominent and highly efficient closed-source model in our evaluation pool.

(3) Tool-ablation models. To decouple "tool availability", "prefixed exemplar context", and "active tool use" as three independent factors, we evaluate representative MLLMs from the first two groups under two ablation regimes. In the *w/o tools* regime, the prompt and decoding hyperparameters are kept unchanged but the tool dispatcher is disabled, so the model must answer directly. In the *prefix* regime, the dispatcher is also disabled, but the prompt is augmented with the $K{=}3$ analogical exemplars (problem statement and reference solution) that the `exemplar_recall` tool would have returned for the target question; this mimics a conventional retrieval-augmented setup where the exemplars are passively read rather than actively retrieved. The two regimes together produce five tool-ablation baselines.

- **Claude-Haiku-4.5 (w/o tools)**: a representative compact closed-source model evaluated without tools, used to verify whether closed-source models in our pool can recover scientific-reasoning performance through internal knowledge alone.
- **GPT-5-mini (w/o tools)**: It evaluates the closed-source versions of the gpt family of closed-source models in the pool, to test whether the general closed-source MLLM can make up for the shortcomings of the missing tools on experimental scientific reasoning tasks.
- **Claude-Haiku-4.5 (prefix)**: the same Claude model as above, but evaluated with $K{=}3$ statically pre-injected exemplars instead of agentic retrieval; comparing this entry against the corresponding *w/o tools* entry quantifies how much performance is recovered when the relevant exemplars are simply handed to the model.
- **Qwen3-VL-30B (prefix)**: a representative large open-source MLLM evaluated under the prefix-RAG regime; together with the Qwen3-VL-8B baselines this row tests whether a $\sim 4\times$ larger backbone benefits from passive exemplar context.
- **Qwen2.5-VL-32B (prefix)**: a 32B open-source MLLM from a different family, evaluated under the prefix-RAG regime as a cross-family check that our *prefix* effects generalize beyond the Qwen3-VL series.

(4) Training baselines. The training group shares MimicAgent's backbone (Qwen3-VL-8B-Instruct) and the same five-tool stack and differs only in the training objective. We align the rollout budget of every training baseline with that of MimicAgent so that any difference in performance reflects the choice of objective rather than the amount of policy interaction.

- **SFT** [34]: pure supervised fine-tuning of the backbone on the same training split using teacher trajectories generated by π_T (`gpt-5.2`); included as the standard imitation-learning baseline.
- **GRPO** [41]: group-relative policy optimization with the same outcome-conditioned reward R_{total} as MimicAgent, but without imitation injection; used to isolate the effect of pure RL on top of our reward.
- **DAPO** [51]: a recent decoupled-clip variant of GRPO that combines a higher clipping ceiling with dynamic sampling; included to compare against a strong RL-only baseline.
- **GSPO** [57]: a sequence-level group-relative policy optimization variant that performs importance sampling and clipping at the sequence level; included as an additional RL-only competitor.
- **SFT+GRPO** [41]: a two-stage pipeline that first SFT-warms up the backbone on teacher trajectories and then runs GRPO on top; this is the strongest RL-only training baseline and the most direct competitor to MimicAgent's adaptive switching.

A.3 Implementation Details

We describe the full implementation of MimicAgent in prose, organized into six paragraphs that cover hardware and software, rollout sampling, GRPO and optimization, the reward and ARIS configuration, the tool service, and the evaluation protocol.

Hardware and software stack. All training is performed on 8 NVIDIA H100 80GB GPUs. Qwen3-VL-8B-Instruct serves as the backbone, SGLang is used as the rollout engine, and FlashAttention-2

is used as the attention backend. We adopt tensor parallelism with TP=2, pipeline parallelism with PP=2, and sequence parallelism, and enable full uniform activation recomputation to stabilize the memory footprint.

Rollout and sampling. At each training step we sample 8 prompts; for each prompt the policy π_θ generates G=8 independent rollouts, yielding a global batch of 64 trajectories. The decoding temperature is fixed at 0.8, with a maximum prompt length of 3,600 tokens, a maximum response length of 7,680 tokens, and an overall context length of 10,240 tokens. Each rollout allows at most 6 agent–tool interaction turns and at most 1 tool call per turn. The SGLang serving engine is configured with a static memory fraction of 0.7 and a maximum concurrency of 160, and the executor timeout for any single tool call is 360 seconds.

GRPO and optimization. GRPO uses a group-relative advantage estimator, with an asymmetric PPO-style clipping range of ϵ_{low}=0.20 and ϵ_{high}=0.28. Both the KL and the entropy coefficients are set to 0. We use the Adam optimizer with (β_1, β_2)=$(0.9, 0.95)$, a weight decay of 0.1, and gradient clipping at norm 1.0. The learning rate is held constant at 1×10^{-6} throughout training.

Reward and ARIS configuration. The total reward is the weighted sum $R_{\text{total}} = 0.40\, R_{\text{answer}} + 0.15\, R_{\text{format}} + 0.25\, R_{\text{tool}} + 0.20\, R_{\text{analogy}}$. The *answer reward* $R_{\text{answer}} \in \{0, 1\}$ is produced by a two-stage judging pipeline: a primary judge `gpt-4.1-mini` returns *correct* ($R_{\text{answer}} = 1$) or *incorrect* ($R_{\text{answer}} = 0$); when its repeated samplings fail to yield a consistent verdict it returns *unknown*, in which case a secondary judge `gemini-3-flash` re-judges, and any residual *unknown* after the secondary stage (about 5%) is counted as incorrect ($R_{\text{answer}} = 0$). The *format reward* $R_{\text{format}} \in \{0, 1\}$ is a schema-only check, independent of the final answer. The *tool-use reward* $R_{\text{tool}} \in [0, 1]$ combines a base score on the trajectory's tool-call quality with a multiplicative outcome-conditioning factor: full credit when *correct*, halved when *unknown*, and reduced to 0.2 times the base when *incorrect*; the precise computation is given below. The *analogical-reasoning reward* $R_{\text{analogy}} = \max(s_{\text{id}}, s_{\text{map}}, s_{\text{trans}})$ is produced by a separate reward model `gemini-3-flash` that scores the three components on a $\{0, 1\}$ scale, and is then composed with the same multiplicative rule as R_{tool}; trajectories that never invoke `exemplar_recall` receive $R_{\text{analogy}} = 0$. Since every term lies in $[0, 1]$, $R_{\text{total}} \in [0, 1]$. The zero-answer indicator used by ARIS is $I_{\text{zero}}(q)$=$\mathbf{1}[\max_i c_i$=$0]$; when $I_{\text{zero}}(q)$=1 we draw an expert trajectory y^* from the teacher policy π_T (`gpt-5.2`) and apply an imitation loss with scaling λ=10.

Computation of R_{tool}. Given a trajectory, let n be the total number of `<tool_call>` blocks emitted (whether or not the JSON parses), and let $m \le n$ count those calls that simultaneously satisfy three conditions: the JSON parses into a valid `(name, arguments)` pair, the tool is successfully executed and returns a non-error result, and the call is not a duplicate of an earlier successful call within the same trajectory in tool name and arguments. We define the base score $\text{base} = m/n \in [0, 1]$, and assign $R_{\text{tool}} = 0$ outright to trajectories with no tool calls. The base is then multiplicatively composed with the answer verdict: R_{tool} equals the full base when the verdict is *correct*, half the base when the verdict is *unknown*, and one fifth of the base when the verdict is *incorrect*. This keeps $R_{\text{tool}} \in [0, 1]$ and proportional to actual tool-call quality in every regime: trajectories with poor tool use receive little reward even when correct, and trajectories that fail still retain a faint signal proportional to their base score, so that the tool-learning channel does not collapse under sparse failure signals during early training.

Tool Service Infrastructure. All five tools are distributed via a unified FastAPI executor service. `exemplar_recall` performs an indexed lookup against the pre-aligned exemplar repository and deterministically returns K=3 companion exercises for any target question; `web_search`, `kg_knowledge`, and `image_crop` call their respective backends; `code_interpreter` executes Python in a sandboxed namespace pre-loaded with `numpy`, `scipy`, `sympy`, `matplotlib`, and `pandas`. All tool calls share an executor timeout of 360 seconds, after which the call returns a `tool_response` error object so that the agent can recover gracefully.

Evaluation protocol. For open benchmarks (SciExplore-Bench, SceMQA), final-answer scoring shares the same dual-judge LLM-as-Judge pipeline used at training time: the primary judge `gpt-4.1-mini` returns *correct* / *incorrect*, falls back to *unknown* when its repeated samplings disagree, and the secondary judge `gemini-3-flash` then re-judges; any residual *unknown* after the secondary stage (about 5%) is counted as incorrect ($R_{\text{answer}} = 0$). For closed MCQ benchmarks (MM-IQ, VisualPuzzles) with single-letter A/B/C/D answers, exact match scoring is used with robust letter extraction regex `<answer>...  </answer>` content. Following [27], we perform a single-pass evaluation on SceMQA. On SciExplore-Bench, we run each experiment five times and report the average performance.

A.4 Tool Implementation Details

This subsection complements the functional description in Section 4.1 with implementation-level provenance: for each tool we briefly state its role inside MimicAgent and which open-source projects, services, or data sources we build on. We deliberately keep this list at the recipe level rather than turning it into a configuration dump; precise endpoints and timeouts already appear in the Tool Service Infrastructure paragraph above.

- `exemplar_recall`**.** The most distinctive tool in our suite, responsible for exemplar-guided analogical reasoning. Each target problem in the curatorial repository is pre-aligned by domain experts with *four* companion exercises in a fixed curatorial order; for any target the tool performs a deterministic indexed lookup against the curated index and returns the default $K{=}3$ of these four companions together with their reference solution traces. Consistent with Section 4.1, the alignment between a target problem and its companions is curatorial, and retrieval is therefore an indexed lookup rather than a similarity-based retrieval.
- `web_search`**.** Open-domain web evidence is served by the `sandbox_serper` backend from the open-source Fathom-DeepResearch project [44], which itself wraps a Serper-indexed search service. We expose it through the same FastAPI executor as the rest of the suite and add a thin layer for authentication, rate limiting, and length-capped responses, so that the tool returns a token-bounded payload usable inside a single rollout turn.
- `kg_knowledge`**.** Structured-knowledge lookup is performed against the AutoSchemaKG framework [6], more specifically against its ATLAS-Wiki sub-graph hosted by a Neo4j-backed server. We wrap this server with a small launcher and expose it as a single tool endpoint, so that an agent query is translated into a multi-hop traversal and the result is condensed into evidence-conclusion pairs in the same JSON envelope as the other tools. Following Section 4.1, `gpt-4.1-mini` is invoked as a bilingual translator and filter on the retrieved evidence-conclusion pairs before they are returned to the agent.
- `image_crop`**.** Region-level visual inspection is implemented in-house, deployed alongside the same sandbox executor as `web_search` for shared image I/O. Given an image and a set of normalised crop coordinates, the tool returns a re-sampled sub-region that is re-injected as a multimodal observation, allowing the agent to read fine-grained visual evidence without re-prompting the entire figure.
- `code_interpreter`**.** A sandboxed Python executor we implemented on top of the same controlled environment shared with `web_search` and `image_crop`. The sandbox pre-loads `numpy`, `scipy`, `sympy`, `matplotlib`, and `pandas`; standard output is captured and length-bounded before being returned as the tool response, so that the agent can move from approximate symbolic reasoning to verifiable numerical computation while staying inside the per-turn budget.

B Limitations and Future Work

At present, SciExplore-Bench focuses on physics, chemistry, and biology, leaving broader scientific coverage for future work. Extending the benchmark and the associated exemplar repository to additional disciplines, such as earth science, materials science, or engineering, would provide a more comprehensive testbed for imitation-driven scientific reasoning. Another limitation is the current modality coverage. Although the benchmark is multimodal, it is still centered on static images. Future work could expand the setting to richer experimental modalities, such as videos, time-series sensor streams, and other dynamic observational data, where procedural imitation may play an even more important role.

C Analysis of SciExplore-Bench

This section provides a detailed analysis of SciExplore-Bench. We first contrast it with existing scientific-reasoning benchmarks along several axes, then characterize its internal composition through knowledge-point coverage, visual content, difficulty distribution, and exemplar repository structure, and finally describe representative problems.

Table 5: Comparison of SciExplore-Bench with existing scientific-reasoning benchmarks. "Exp." indicates whether the benchmark explicitly focuses on experimental scientific inquiry; "Bilingual" indicates whether both Chinese and English evaluation splits are provided; "Train" indicates whether a paired training split is available.

Benchmark	Year	Size	Subjects	Multimodal	Format	Exp.	Bilingual	Train
MMLU [19]	2020	15,908	57 disciplines (incl. STEM)	–	MCQ	–	EN	–
ScienceQA [29]	2022	21,208	natural / social science	Yes	MCQ	–	EN	Yes
C-Eval [21]	2023	13,948	52 disciplines (incl. STEM)	–	MCQ	–	ZH	–
CMMLU [25]	2023	11,528	67 disciplines (incl. STEM)	–	MCQ	–	ZH	–
SciBench [49]	2023	695	basic science / math	–	open-ended	–	EN	–
GPQA [38]	2023	448	graduate-level science	–	MCQ	–	EN	–
GAOKAO-Bench [56]	2023	2,811	multi-discipline (Chinese gaokao)	Part	MCQ + open	–	ZH	–
TheoremQA [11]	2023	800	theorem-driven QA	–	open-ended	–	EN	–
SciEval [46]	2023	18,000+	multi-discipline	–	MCQ + open	Part	EN	–
OlympiadBench [18]	2024	8,476	olympiad-level science	Yes	open-ended	–	ZH+EN	–
MMMU [53]	2024	11,500	6 disciplines	Yes	MCQ + open	–	EN	–
MathVista [28]	2024	6,141	visual mathematical reasoning	Yes	MCQ + open	–	EN	–
MMMU-Pro [54]	2024	≈3,500	6 disciplines (enhanced)	Yes	MCQ	–	EN	–
LAB-Bench [23]	2024	2,400+	research assistance	Part	MCQ	Part	EN	–
ScienceAgentBench [13]	2024	102	research-style data analysis	–	code generation	Part	EN	–
SciCode [48]	2024	338	research-style coding	–	code generation	–	EN	–
HLE [36]	2025	2,500	frontier multi-discipline	Yes	MCQ + short ans.	–	EN	–
MMSciBench [50]	2025	≈3,000	multi-discipline science	Yes	MCQ + open	–	ZH	–
SciExplore-Bench (Ours)	**2026**	**6,480**	**Phy / Chem / Bio (experimental)**	**Yes**	**open-ended inquiry**	**Yes**	**ZH+EN**	**Yes**

C.1 Comparison with Existing Benchmarks

Table 5 compares SciExplore-Bench with 18 representative scientific-reasoning benchmarks along year, size, subject coverage, multimodality, question format, experimental focus, language coverage, and the availability of a training split. To the best of our knowledge, SciExplore-Bench is the first multimodal scientific benchmark that simultaneously satisfies four properties: it uses fully open-ended scientific-inquiry question formats, focuses on experimental scientific reasoning, supports bilingual evaluation, and ships with a paired training split.

Four observations follow from Table 5, with Figure 5 providing a visual summary along six feature dimensions. *(i) Format.* Most existing benchmarks rely on multiple-choice questions, code generation, or theorem-style derivations, whereas SciExplore-Bench is fully open-ended at the level of scientific inquiry, requiring models to produce complete analyses rather than option letters. *(ii) Experimental focus.* Even when prior benchmarks cover science as a domain, they rarely target the cluster of competencies tied to experimental practice—experimental design, variable control, evidence interpretation, and mechanism inference; SciExplore-Bench is one of the few benchmarks built around these competencies as the explicit evaluation target. *(iii) Bilingual + multimodal coverage.* Bilingual scientific benchmarks that also cover image-based content are scarce: OlympiadBench is bilingual and multimodal but focuses on olympiad-style derivations, while MMSciBench is Chinese-only. *(iv) Training split.* Most benchmarks ship only a test split, which prevents systematic study of how to acquire the targeted ability; SciExplore-Bench pairs 480 bilingual evaluation problems with 6,000 rewritten training problems that share the rewriting pipeline, supporting supervised, reinforcement, and adaptive training paradigms.

C.2 Knowledge Coverage Analysis

Our evaluation split covers 187 distinct knowledge-point tags, spanning physics (kinematic measurement, electromagnetic exploration, optical measurement), chemistry (reaction kinetics, substance identification, ionic-reaction analysis, and experimental scheme evaluation), and biology (microbial culture experiments, enzymatic activity exploration, cellular process observation, population-density estimation), among others. We characterize the coverage with two complementary views. Figure 6 visualizes the 187 knowledge points in a t-SNE projection of their character-level TF-IDF embeddings, with point size reflecting question frequency and color reflecting subject. Figure 7 shows a subject-coloured word cloud over the same tags. Both views reveal multiple well-separated semantic clusters and a high degree of tag diversity, indicating that SciExplore-Bench is not dominated by a few narrow topics but instead covers a wide and diverse set of experiment-centered reasoning scenarios.

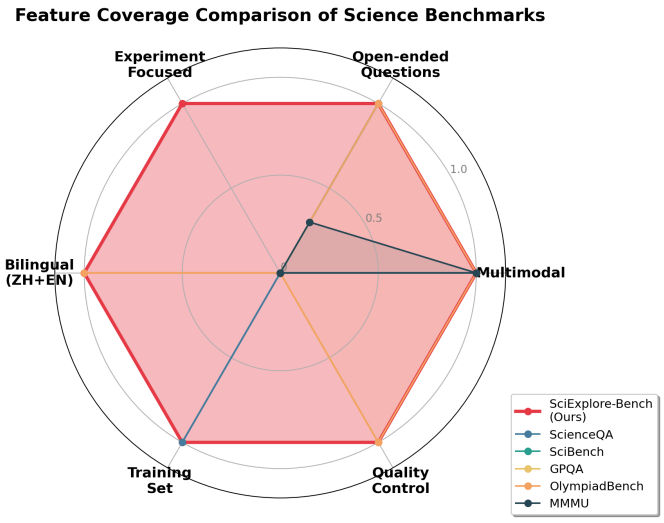


Figure 5: Six-dimensional feature comparison between SciExplore-Bench and representative scientific reasoning benchmarks. The dimensions are: *multimodal coverage*, *open-ended format*, *experimental focus*, *bilingual evaluation*, *training-split availability*, and *automated quality control*. SciExplore-Bench is the only benchmark that fully covers all six dimensions.

C.3 Visual Content Analysis

Approximately 66.2% (159 out of 240) of the evaluation problems in the test split are image-bearing (Table 3). These images cover three major categories—experimental apparatus diagrams, data plots and charts, and microscopic / cell illustrations—plus a residual "other" category. Figure 8 reports the approximate proportion of each category over the 206 images attached to those image-bearing problems. Experimental apparatus diagrams dominate the distribution ($\sim 56\%$), followed by data charts ($\sim 29\%$), microscopic / cell illustrations ($\sim 8\%$), and other diagrams ($\sim 6\%$), which reflects the experimentally grounded nature of the visual evidence.
We further compute the resolution distribution of these 206 images. The shortest-side resolution has minimum / median / 95th percentile / maximum equal to 34 / 293 / $1{,}123$ / $1{,}881$ pixels respectively, while the longest-side counterparts are 90 / 608 / $2{,}683$ / $3{,}996$ pixels (Figure 8, panel b). This long-tailed distribution spans textbook line drawings, schematic figures, and high-resolution observation images, and stresses the agent's ability to both localize task-relevant local regions through `image_crop` and to extract numerical evidence from densely populated data plots.

C.4 Difficulty Distribution

We follow the difficulty levels (easy / medium / hard) annotated in the source corpora. Table 6 reports the per-tier statistics of the test and training splits, and Figure 9 provides the corresponding visualization. The Chinese training pool ($3{,}000$ problems) has difficulty proportions $26.4\%/56.0\%/17.6\%$, while the Chinese test set (240 problems) has proportions $29.6\%/53.3\%/17.1\%$. The two distributions agree closely in the hard tier and differ by roughly 3 percentage points in the easy and medium tiers, so no systematic difficulty mismatch is introduced between training and evaluation. Figure 9(b) provides a per-subject view, showing that the per-subject training pool is highly asymmetric—chemistry ($1{,}196$) and biology ($1{,}231$) dominate physics (573)—whereas the test set forces a uniform 80 / 80 / 80 split. The English version is independently collected following the same pipeline and per-subject sizes are matched to the Chinese split by stratified sampling, so the English split inherits the same per-subject and difficulty proportions.

C.5 Exemplar Repository Statistics

A defining feature of SciExplore-Bench is its pre-prepared curatorial exemplar repository, in which each target problem is paired with *four* companion exercises by domain experts and `exemplar_recall` returns the default $K{=}3$ of these four companions in fixed curatorial order.

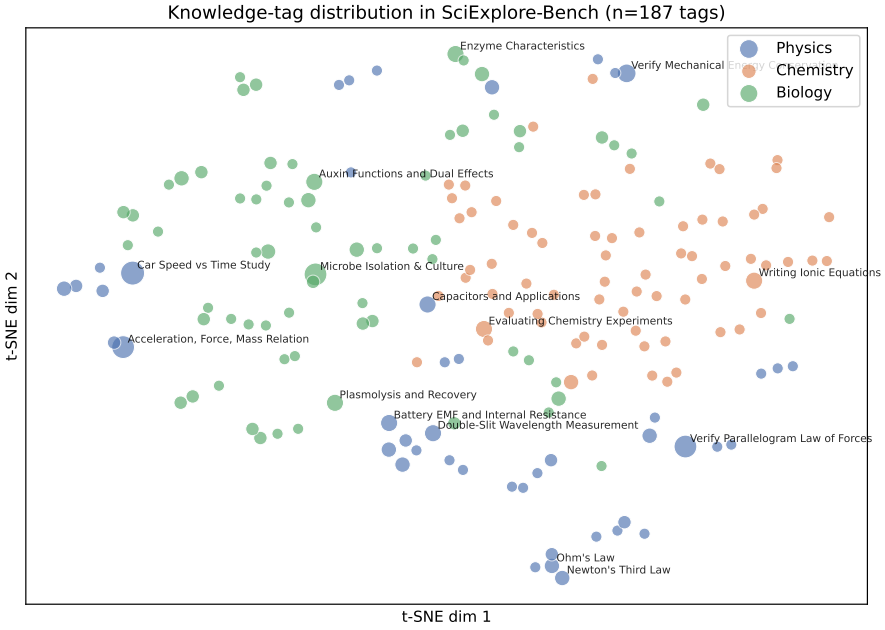


Figure 6: Semantic distribution of the 187 knowledge-point tags in SciExplore-Bench. Each point corresponds to one knowledge tag, embedded by character-level TF-IDF features over its English label and projected to two dimensions via t-SNE. The size of a point reflects the number of questions covered by that tag, and the colour reflects the subject (physics / chemistry / biology); the top 15 most frequent tags are annotated in-place.

Table 6: Difficulty distribution of SciExplore-Bench (Chinese split shown; the English split is independently collected with per-subject sizes matched by stratified sampling, and inherits the same distribution). The training column refers to the full Chinese training pool of 3,000 problems and the test column refers to the 240-problem Chinese test set used in this paper.

Difficulty	**Test #**	**Test %**	**Train #**	**Train %**
Easy	71	29.6%	792	26.4%
Medium	128	53.3%	1,679	56.0%
Hard	41	17.1%	529	17.6%
Total	240	100%	3,000	100%

To evaluate the semantic quality of the companions returned by `exemplar_recall`, we use the Qwen3-VL-8B-Embedding model [26] as a third-party text encoder and sample 100 targets per subject (n=300 in total). We report two analyses (Figure 10): (a) the cosine-similarity distribution between each target embedding and the embeddings of its three returned companions, and (b) the distribution of shared knowledge-point tags between each target and its companions.

Figure 10(a) shows the cosine-similarity distributions of the three returned companions, sorted post-hoc by similarity within each query for visualization. All three distributions concentrate at the high end: rank-1 has median 0.824 and mean 0.818, rank-2 has mean 0.795, and rank-3 has mean 0.781, with the lowest observed similarity around 0.46. The vast majority of targets are paired with companions that are also semantically close in the Qwen3-VL-8B-Embedding space.

Figure 10(b) reports the distribution of shared knowledge-point tags between each target and its three returned companions: 75.7% of the targets share at least one tag with their companions (163 share exactly one, 47 share two, 17 share three or more), and only 73 (24.3%) share none. Together, the embedding-similarity and tag-overlap signals show that the companions returned by `exemplar_recall` are consistently close to their targets in both semantic-embedding and topical-tag senses, providing the substrate for the procedural transfer behaviour that MimicAgent learns to exploit.

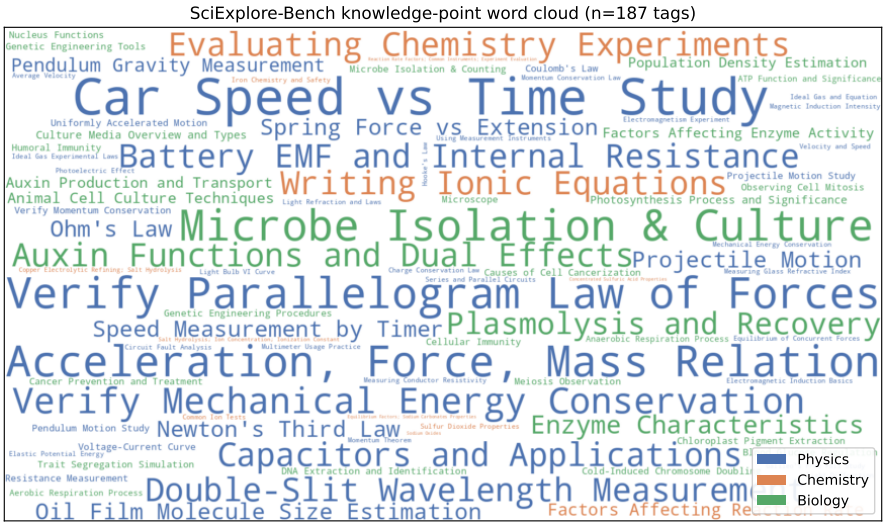
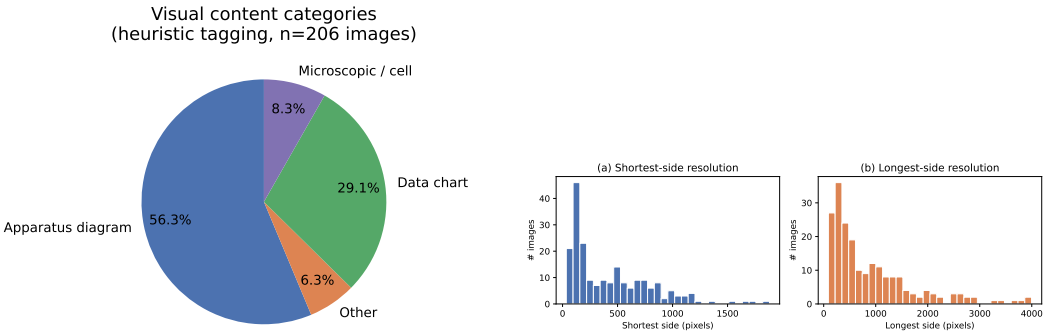


Figure 7: Subject-coloured word cloud over the 187 knowledge-point tags of SciExplore-Bench. Tag size is proportional to the number of questions covered by the tag; colour denotes the subject (physics, chemistry, biology). The diversity of high-frequency tags across all three subjects further illustrates that SciExplore-Bench covers a broad range of experiment-centered topics rather than a narrow slice.



(a) Visual content categories.

(b) Image resolution histogram (shortest / longest side, n=206).

Figure 8: (a) Distribution of visual content categories among the 159 image-bearing test problems (heuristic tagging based on question text and knowledge tags). (b) Resolution distribution of the 206 images attached to those problems, plotted separately for the shortest and longest side.

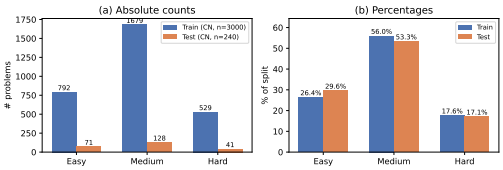
C.6 Representative Examples

We list six representative SciExplore-Bench problems in Table 7, two per subject and collectively spanning four cognitive tasks (experimental design, variable control, data analysis, and mechanism inference). For each problem we give a one-sentence English summary of the target question and a one-sentence summary of the reference transfer skeleton from its analogical exemplar; together they illustrate the open-ended scientific-inquiry style of the benchmark.

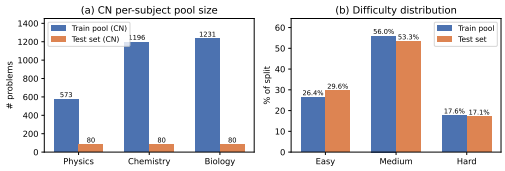
D Additional Experiments

D.1 Algorithm

We summarize the prompt-level switching logic of ARIS as the pseudocode in Algorithm 1, so that the per-prompt mutual exclusion and the convex-combination behaviour at the batch level can be grasped at a glance.



(a) Test set vs. training pool.



(b) Per-subject pool sizes and difficulty ratios.

Figure 9: Difficulty distribution of the SciExplore-Bench training and test splits (Chinese; the English split mirrors the Chinese split via stratified sampling on independently collected English sources). Subject-wise sampling enforces a uniform 80 / 80 / 80 test split despite the asymmetric training pool, while the difficulty ratios remain close between training and evaluation.

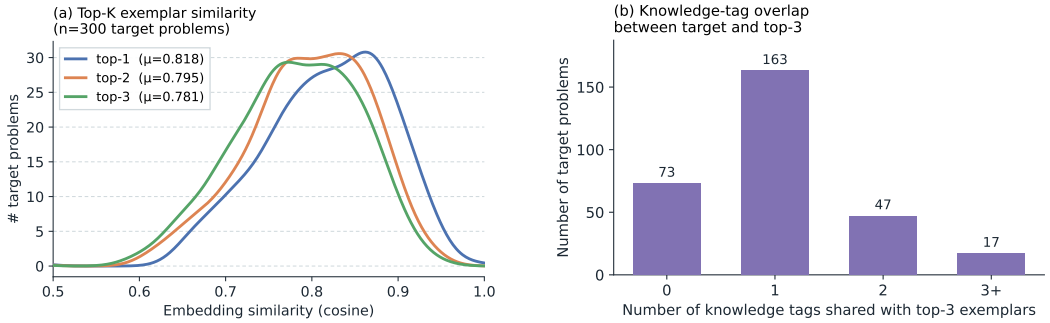


Figure 10: Quality evaluation of `exemplar_recall` retrieval results, computed by probing the Qwen3-VL-8B-Embedding model [26] for n=300 target problems sampled from the three training subjects. (a) Cosine-similarity distribution between each target and the three companions returned by `exemplar_recall` under the default K=3, sorted post-hoc by similarity within each query for visualization (one curve per rank). (b) Histogram of the number of knowledge-point tags shared between each target problem and its three returned companions.

ARIS performs exactly one weighted policy update per prompt. When the on-policy group contains at least one correct rollout, ARIS reduces to standard GRPO; when all rollouts fail, the GRPO update is replaced by an imitation update on a teacher-generated trajectory. Although mutually exclusive at the prompt level, the batch-level expected loss remains a weighted mixture whose mixture coefficient $p_{\text{zero}}^{(k)}$ decays automatically as the student improves.

D.2 Reward Dynamics

We record the evolution of the four reward dimensions during training in Figure 11(a), together with a side-by-side comparison of mean rewards on correct versus incorrect samples for the two outcome-conditioned process dimensions in Figure 11(b).

Figure 11(a) reveals three training phases. *Early* (first $\sim 13\%$ of training): R_{format} saturates near 0.7 before any other dimension improves, indicating that the agent first learns to wrap its outputs into `<think>` / `<tool_call>` / `<answer>` blocks while most answers remain incorrect. *Mid* ($\sim$ 13–69%): R_{tool} and R_{analogy} rise together, indicating simultaneous improvement of tool-call quality and exemplar-driven procedural transfer. *Late* (beyond $\sim 69\%$): R_{analogy} exhibits a sharp jump to roughly 0.9, corresponding to the moment when the agent internalizes "when to retrieve and how to use the retrieved exemplar", and R_{total} rises from ~ 0.2 to ~ 0.6.

Figure 11(b) provides a direct empirical verification of outcome-conditioned discriminability. On the two process dimensions that are outcome-conditioned, correct samples receive substantially higher mean rewards than incorrect ones: R_{tool} scores 0.816 versus 0.637 (gap 0.179), and R_{analogy} scores 0.517 versus 0.208 (gap 0.309). The ordering $\Delta R_{\text{analogy}} > \Delta R_{\text{tool}}$ matches the design intuition that analogical reasoning is the most discriminative process-level behaviour of MimicAgent, and reaffirms the Section 4 motivation for outcome-conditioned modulation of the process rewards. R_{format} is not outcome-conditioned and is therefore omitted from this comparison.

Table 7: Six representative SciExplore-Bench problems, two per subject and collectively covering experimental design, variable control, data analysis, and mechanism inference.

#	Subject	Cognitive task	Target problem	Reference transfer skeleton
1	Physics	data analysis	Verify momentum conservation on an air track when the gates use *non-uniform* flag widths and the post-collision gate is *far* from the impact point, so that the recorded gate speed must be corrected for friction-induced deceleration.	Replace the unobservable instantaneous speed by the proxy d/T; unify sign conventions; correct the post-impact proxy via $v_{C0}^2 = (d_C/T_C)^2 + 2a_f L$ before applying the conservation relation.
2	Physics	mechanism inference	Compute the net thrust on a spacecraft when a fixed deflector behind an ion thruster either absorbs the plume, deflects it by 90°, or reflects it by 180°, given mass flow q and exhaust speed v.	Close the momentum-flux balance between thruster and deflector; superpose action-and-reaction forces; convert the qualitative recoil argument into the quantitative net force in each boundary case.
3	Chemistry	experimental design	Rank four candidate metal oxides as H_2O_2 decomposition catalysts, verify that each oxide is recovered chemically unchanged, and identify the side reaction that explains an anomalous solid colour change for PbO_2.	Parallel screening across candidates; activity ordering by reaction time; recovery-and-recheck protocol verifies that the catalyst is unchanged; an anomalous product localises a redox side reaction.
4	Chemistry	mechanism inference	Diagnose why $Fe(SCN)_3$ colorimetry of Fe^{3+} in industrial waste fails when trace TBP is present (water phase becomes colourless while the floating organic layer turns deep red), and design a corrected detection scheme.	Anomalous indicator signal $\rightarrow$ pose two competing hypotheses (chemical consumption vs. phase-transfer extraction) $\rightarrow$ separate them by a controlled reduction-reagent test and an independent extraction test, then localise the true cause.
5	Biology	experimental design	Test whether the vagus nerve regulates LPS-induced TNF-α release in mice, including a sham-surgery control and a follow-up acetylcholine rescue experiment.	Lesion an upstream regulatory structure + apply a downstream functional stimulus + read the output; control surgical confounds with a sham group; close the loop by rescuing the lesioned phenotype with the missing transmitter.
6	Biology	variable control	Determine which of (jasmonate inducer, phenylalanine precursor, agitated aeration) are individually necessary for taxol production in *Taxus* suspension cultures, after first depleting endogenous reserves by serial passage.	Multi-factor necessity test; deplete endogenous reserves to remove the confound; use single-factor knockout groups to attribute the contribution of each factor to the final yield.

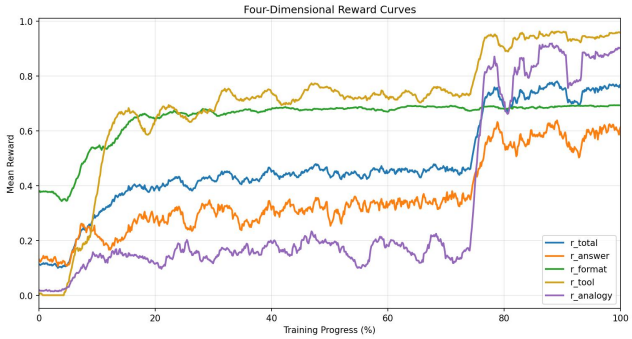
Algorithm 1 ARIS update for a single prompt q.

Require: prompt q, group size G, teacher policy π_T, tool set $\mathcal{T}$, format $\mathcal{F}$, scaling λ, learning rate η
Ensure: updated policy parameter θ

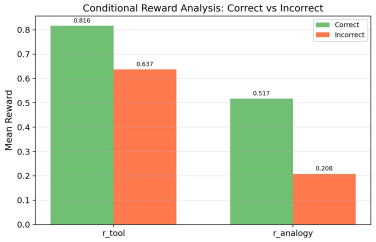
Sample G rollouts $\{o_1, \ldots, o_G\} \sim \pi_\theta(\cdot \mid q; \mathcal{T}, \mathcal{F})$
for $i = 1, \ldots, G$ **do**
  $c_i \leftarrow \text{Judge}(o_i)$ ▷ answer correctness
end for
$I_{\text{zero}} \leftarrow \mathbf{1}[\max_i c_i = 0]$
if $I_{\text{zero}} = 0$ **then** ▷ at least one rollout correct
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{GRPO}}(q)$ ▷ group-relative advantage
else ▷ all rollouts failed
  $y^\star \sim \pi_T(\cdot \mid q; \mathcal{T}, \mathcal{F})$ ▷ online expert trajectory
  $\mathcal{L} \leftarrow \lambda \cdot \mathcal{L}_{\text{imit}}(q, y^\star)$ ▷ token-masked imitation
end if
$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$

D.3 Inference Efficiency Dynamics

Figure 12 reports the cost per correct answer over training. The token cost decreases from roughly $15{,}000$ to roughly $5{,}000$ tokens (about a $3\times$ reduction) and the tool-call count decreases from about 30 to about 5 per correct answer (about a $6\times$ reduction). Both curves are plotted against the normalized training progress (0–100%) and decrease in lock-step with the rising correct rate: as training proceeds, the agent reaches correct answers via shorter trajectories rather than longer trial-and-error chains. This efficiency gain is not in tension with the response-length growth reported in Section 5.3 (mean response length rises from $\sim 1{,}500$ to $\sim 3{,}000$ tokens): the latter measures the mean length of *a*



(a) Four-dimensional reward curves versus training progress.



(b) Mean reward of correct vs. incorrect samples on the two outcome-conditioned process dimensions.

Figure 11: Reward dynamics during ARIS training. (a) Mean reward of R_{total}, R_{answer}, R_{format}, R_{tool}, and R_{analogy} versus training progress; the late jump in R_{analogy} marks the onset of internalized exemplar-driven imitation. (b) Conditional reward analysis on the two outcome-conditioned dimensions (R_{tool} and R_{analogy}): correct samples obtain higher mean rewards than incorrect ones, with the R_{analogy} gap being the largest. R_{format}, which is not outcome-conditioned, is omitted from this panel.

single rollout, whereas the former measures the cumulative cost (across rollouts and tool calls) needed to obtain *one correct answer*. The former reflects the agent's ability to engage in deeper multi-step reasoning, while the latter reflects that this reasoning becomes much more likely to land on the correct answer rather than be wasted on failed exploration.

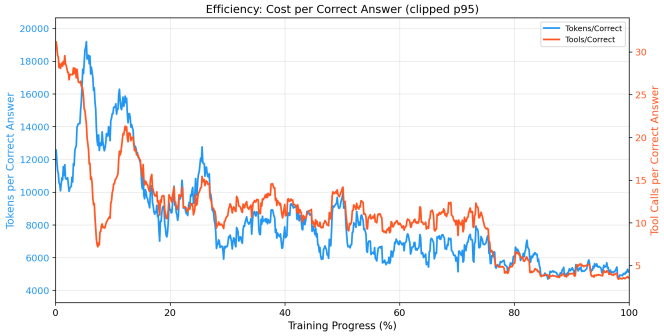


Figure 12: Inference efficiency during ARIS training, plotted against the normalized training progress (0–100%). The blue curve (left axis) shows tokens consumed per correct answer; the orange curve (right axis) shows tool calls invoked per correct answer. Both decrease by roughly 3–6$\times$ over training, indicating that the agent learns to reach correct conclusions through shorter, more targeted trajectories.

D.4 Exemplar Injection Size Ablation

Each target in the curatorial repository is paired with *four* companion exercises by domain experts, and `exemplar_recall` returns the first $K{=}3$ of them by default. To remove any confound from the agentic retrieval policy, we take the trained MimicAgent and evaluate it under the *prefix* regime defined in Appendix A.2 – the agentic `exemplar_recall` call is replaced by a static prefix of K pre-aligned companions at the end of the prompt, with no tool dispatch – and sweep K from 1 to 4 on a 240-problem EN evaluation subset (80 per subject). As shown in Figure 13, $K{=}3$ attains the highest accuracy at 67.08%, more than nine percentage points above the next best value among $K \in \{1, 2, 4\}$. Both fewer ($K{\leq}2$) and the maximal ($K{=}4$) prefix exemplars degrade accuracy into the 52–58% range: too few companions do not span enough transferable schemas to support analogical transfer, while exhausting all four companions introduces lower-relevance exemplars that dilute the model's attention over the target problem and crowd out the budget left for tool-augmented reasoning. The $K{=}3$ peak validates the default value used everywhere else in the paper.

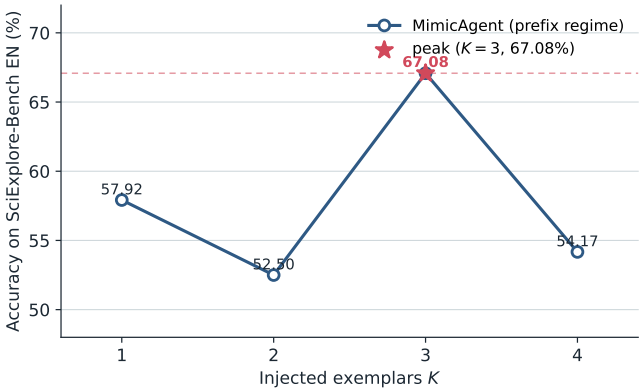


Figure 13: Ablation on the number of injected exemplars K for `exemplar_recall`. Each target in the curatorial repository is paired with four companion exercises and `exemplar_recall` returns the first $K{=}3$ by default. We take the trained MimicAgent and replace its agentic `exemplar_recall` call with the *prefix* regime defined in Appendix A.2: K pre-aligned companions (problem statement and reference solution) are statically prepended to the prompt without going through the tool dispatcher. Accuracy (%) is reported on the EN evaluation subset of SciExplore-Bench (240 problems, 80 per subject). $K{=}3$ (red star) matches the default of `exemplar_recall` and outperforms every other $K \in \{1, 2, 4\}$ by at least nine percentage points.

D.5 Case Study

We provide three end-to-end case studies that trace MimicAgent's full reasoning pipeline on SciExplore-Bench problems, from tool-augmented evidence gathering to exemplar retrieval, structural alignment, and a verifiable final answer. The three cases come from physics (signal-based mechanism inference), chemistry (control-difference reaction-enthalpy diagnosis), and biology (multi-factor necessity testing), and together illustrate how the same four-step recipe – (i) build scientific context with tool calls, (ii) retrieve a structurally analogous exemplar via `exemplar_recall`, (iii) abstract a transferable solution schema, and (iv) align variable roles to produce the target answer – adapts to qualitatively different scientific tasks.

(1) Eddy-current damping tower (physics, signal-based mechanism inference). Figure 14 traces MimicAgent's full trajectory on a contactless electromagnetic damping problem. Two long tubes (A, B) are wound on the outside with detection coils placed every distance h. As a magnetised capsule falls through tube A, the first three induced-EMF pulses grow in peak amplitude and shrink in inter-pulse interval; from the fourth pulse onward both peak and interval stay constant. In tube B the pulse peaks grow monotonically and the intervals shrink monotonically without saturating. The agent must (a) decide which tube hosts a closed copper-ring array and which a slotted (open-circuit) array, (b) reconstruct the motion of the capsule in each tube, and (c) propose two engineering changes that reduce the steady falling speed inside the damped tube without altering the capsule. MimicAgent first uses two `web_search` calls to anchor the verifiable fact that "closed conducting loops sustain a continuous induced current while slotted rings cannot close a circuit", and then a `kg_knowledge` call to retrieve Faraday's law $\varepsilon \propto -\mathrm{d}\Phi/\mathrm{d}t$ and Lenz's law as the canonical mechanism template. A single `exemplar_recall` call returns a structurally analogous precedent in which a magnet falls through an aluminium tube versus a glass tube and the coil-current time series is used to back-out terminal velocity in the conducting medium and free fall in the insulating one. Rather than reusing the precedent's numerical answer, MimicAgent abstracts a transferable schema – *paired-medium control + temporal-signal feature identification + feedback-damping mechanism diagnosis* – and writes the variable–role alignment explicitly inside <THINK>: *aluminium tube* $\rightarrow$ tube A (closed-ring array), *glass tube* $\rightarrow$ tube B (slotted array), *coil-current time series* $\rightarrow$ detection-coil pulse train, *eddy-current damping* $\rightarrow$ ampère force on the induced ring current, *terminal velocity* $\rightarrow$ the saturated peak/interval signature observed from the fourth pulse onward, and *free fall* $\rightarrow$ the unsaturated, monotonically growing pulse train. The four schema items are then applied to the four sub-tasks: (i) the late-stage saturation in tube A maps onto the precedent's "terminal velocity in aluminium" criterion, fixing tube A as the closed array and tube B as the slotted one; (ii) using $\varepsilon \propto v$, the rising flux-change rate drives

a rising induced current and a rising ampère drag, instantiating the feedback-damping branch of the schema; (iii) the motion phase is therefore reconstructed as a decelerating-acceleration phase during pulses 1–3 followed by force-balanced cruise from pulse 4 onward in tube A, and a uniform free-fall trajectory in tube B; (iv) because the schema's last item states that drag is monotonic in induced current, MimicAgent recommends two complementary changes – reducing the ring resistance (more current per unit $\mathrm{d}\Phi/\mathrm{d}t$) and strengthening the magnet's moment (larger $\mathrm{d}\Phi/\mathrm{d}t$ per unit displacement) – both of which lower the equilibrium speed at which gravity balances the magnetic drag. Notably, the precedent's specific quantitative conclusions (e.g. "terminal velocity reached within 1.2 m") never appear in the target solution; only the four-item schema does.

(2) CO_2 chemisorption on amine-functionalised silica (chemistry, control-difference reaction-enthalpy diagnosis). Figure 15 traces MimicAgent's full trajectory on an amine-functionalised mesoporous SiO_2 (Amine-SiO_2) sorbent designed to capture CO_2. In an adiabatic calorimeter of fixed total heat capacity C, equal masses of the sorbent receive 1 mol of CO_2 and 1 mol of N_2 in turn, producing temperature rises ΔT_1 and ΔT_2 respectively; N_2 has a similar molecular size to CO_2 but is chemically inert toward amines. The agent must (1) design a control experiment that subtracts physical-adsorption interference from the chemical-reaction heat, (2) write a closed-form expression for the molar reaction enthalpy ΔH, and (3) explain why repeated high-temperature regeneration cycles leave ΔT_2 unchanged but cause ΔT_1 to drop substantially. MimicAgent first uses two `web_search` calls to anchor two verifiable facts – mesoporous SiO_2 contributes only van der Waals physical adsorption, and primary amines react with CO_2 to form a carbamate – and then a `kg_knowledge` call to retrieve the canonical calorimetric relation $q = C\,\Delta T$. A single `exemplar_recall` call returns a structurally analogous precedent in which a water blank is used as a control to subtract the physical dissolution heat of $NaHCO_3$/Na_2CO_3, leaving only the carbonate–HCl reaction enthalpy and its sign. Rather than reusing the precedent's specific numerical conclusions, MimicAgent abstracts a transferable schema – *target signal contains background contribution → select a blank control with the same physical channel and a zero chemical channel → subtract to isolate the chemical heat → assign ΔH from the sign of the differential heat → diagnose material failure by tracking control vs. target across cycles* – and writes the variable–role alignment explicitly inside <THINK>: *solid dissolution heat* → gas physical adsorption heat inside SiO_2 pores, *carbonate–HCl reaction heat* → chemical reaction heat between CO_2 and surface amines, *water blank* → equimolar N_2 control gas, *pre/post-mixing temperature change* → calorimetric temperature rise ΔT. The five schema items are then applied to the three sub-tasks: (i) for the control design, the schema demands a blank that matches the physical channel and zeroes the chemical channel, and N_2 – size-matched to CO_2 but inert toward amines – is the exact analogue of the water blank in the precedent; (ii) for the formula, the "subtract to isolate" instruction yields $Q_{\mathrm{phys}} = C\Delta T_2$, $Q_{\mathrm{total}} = C\Delta T_1$, hence $Q_{\mathrm{rxn}} = C(\Delta T_1 - \Delta T_2)$, and the "assign ΔH from the sign" instruction gives $\Delta H = -C(\Delta T_1 - \Delta T_2)$ since $\Delta T_1 > \Delta T_2$ certifies the reaction as exothermic; (iii) for the deactivation diagnosis, the schema's last item asks MimicAgent to track the control and target curves separately across cycles – ΔT_2 being invariant pins the SiO_2 pore structure as intact, while the drop in ΔT_1 localises the failure to the chemical channel, which the carbamate-formation fact retrieved earlier directly attributes to thermal decomposition or volatilisation of amine sites during high-temperature regeneration. The precedent's water–carbonate numbers and chemical equation never appear in the target solution; only the differencing-and-diagnosis schema is reused.

(3) Paclitaxel biosynthesis (biology, multi-factor necessity testing). Figure 16 traces MimicAgent's full trajectory on a *Taxus* suspension-culture problem with four parallel groups – A: basal medium + MeJA + phenylalanine + agitation; B: minus MeJA; C: minus phenylalanine; D: static (no agitation) – in which the cells are passaged three times in inducer- and precursor-free basal medium before the formal trial. The agent must (1) explain the purpose of the three-pass subculture, (2) explain why "equal cell aliquots in multiple flasks" is part of the design, (3) given that group A yields significantly more paclitaxel than B, C, and D, derive what each pairwise contrast (A–B, A–C, A–D) implies about the necessity of the corresponding factor, and (4) explain why static culture (group D) produces both slow cell growth and low paclitaxel yield. MimicAgent first uses two `web_search` calls to anchor the verifiable facts that MeJA is a canonical elicitor of plant secondary metabolism and that phenylalanine is the upstream precursor of taxane biosynthesis, and a `kg_knowledge` call to retrieve the agitation → dissolved-oxygen → aerobic-respiration → ATP supply pathway. A single `exemplar_recall` call returns a structurally analogous precedent in which the endosperm of wheat seeds is stripped out, after which Mg / light / warmth are individually withdrawn to identify the conditions necessary for chlorophyll synthesis – endosperm removal serves as an *endogenous-reserve depletion* step that

prevents the seed's stored Mg/energy from masking the single-factor knock-outs, while seedling colour reads out each condition's necessity. Rather than reusing this source's answer, MimicAgent abstracts a transferable schema – *multi-factor necessity testing + endogenous-reserve depletion to remove confounds + single-factor knock-out comparison + pathway-level mechanism diagnosis when one group's result is qualitatively different* – and writes the variable–role alignment explicitly inside <THINK>: *chlorophyll* → paclitaxel, *Mg / light / warmth* → MeJA / phenylalanine / agitated aeration, *endosperm removal* → three passes of subculture in unsupplemented medium, *seedling colour* → paclitaxel yield. The four schema items are then applied to the four sub-questions: (i) the "endogenous-reserve depletion" item identifies the three-pass subculture as an internal-MeJA / internal-phenylalanine wash-out, without which the residual endogenous pool would let group B or group C still produce paclitaxel and break the single-variable causality; (ii) the "single-variable design" principle re-frames "equal cell aliquots" as control of irrelevant variables and "multiple flasks" as biological replication for noise reduction; (iii) the "single-factor knock-out" item turns the three pairwise contrasts into three independent necessity statements – A–B (only MeJA differs) certifies MeJA, A–C (only phenylalanine differs) certifies the precursor, A–D (only agitation differs) certifies the oxygen-supply factor; (iv) because group D is qualitatively different (both growth *and* yield drop), the schema's last item triggers a pathway-level explanation using the kg-retrieved chain, yielding "static culture → low dissolved oxygen → impaired aerobic respiration → reduced ATP → simultaneous suppression of cell division and secondary-metabolite biosynthesis". The precedent's chlorophyll/Mg/seedling-colour content never enters the target solution; only the depletion-plus-knock-out-plus-pathway schema is reused, and the four resulting answers match the reference key on every sub-item.

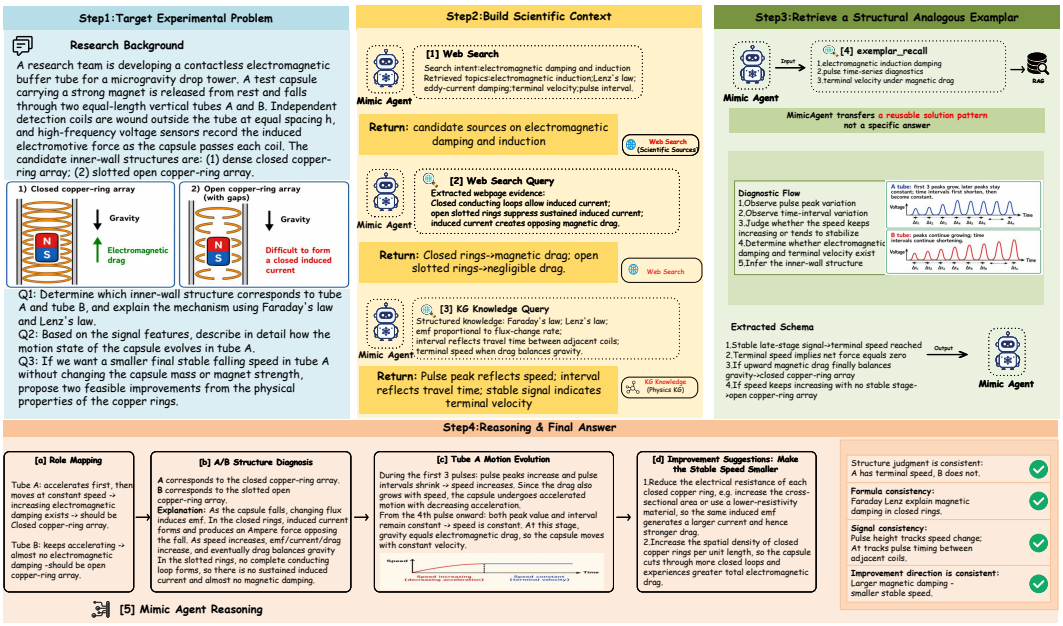


Figure 14: Case study (physics): MimicAgent diagnoses an electromagnetic damping tower from coil pulse-train signals. The four panels follow the agent's actual trajectory, from target-problem inspection, through tool-augmented context building (`web_search` and `kg_knowledge`) and exemplar retrieval (`exemplar_recall` returning the magnet-through-aluminium-vs-glass exemplar), to schema extraction, variable-role alignment, and the final reasoning that yields the structural identification, motion-phase reconstruction, and improvement suggestions.

E Prompt And Consistency Analysis

E.1 Prompt Analysis

We rely on four categories of prompts across our pipeline: *(i)* the answer-judge prompts used in the LLM-as-Judge pipeline that produces R_{answer}; *(ii)* the analogical-reasoning reward prompts

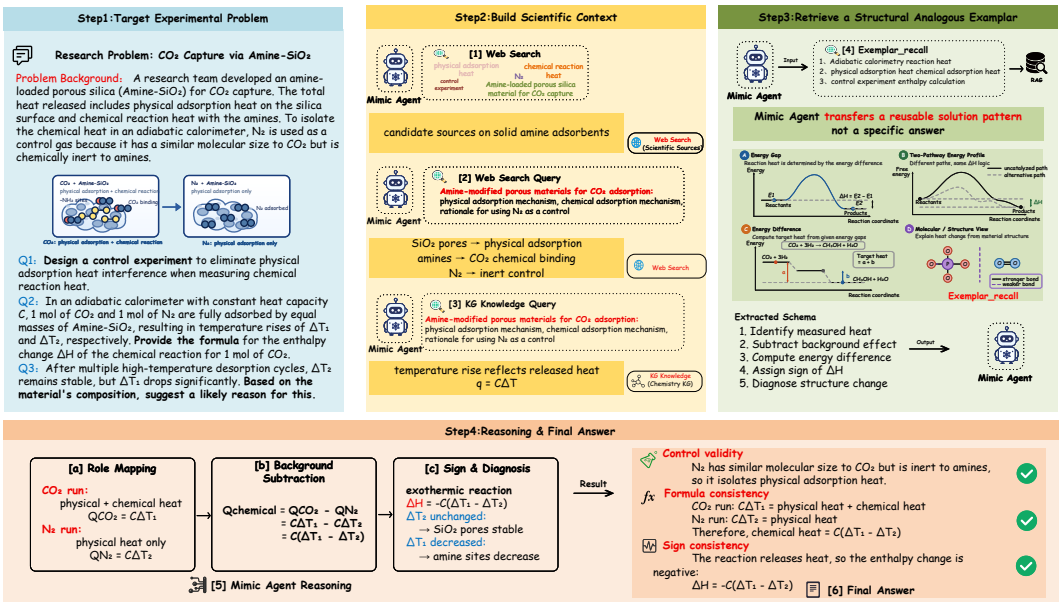


Figure 15: Case study (chemistry): MimicAgent diagnoses CO_2 chemisorption heat on an amine-functionalised mesoporous SiO_2 sorbent. The trajectory shows tool-augmented background building (mesoporous SiO_2 as physical-adsorption-only via `web_search`; carbamate-formation pathway via `kg_knowledge`), exemplar retrieval of a structurally analogous control-experiment / reaction-enthalpy precedent through `exemplar_recall`, schema extraction (*paired control-experiment differencing + enthalpy-sign assignment from the differential heat*), and the final reasoning that separates physical from chemical contributions, assigns $\Delta H_{\mathrm{rxn}} < 0$, and explains the cycle-by-cycle deactivation.

used to score $(s_{\mathrm{id}}, s_{\mathrm{map}}, s_{\mathrm{trans}})$; *(iii)* the teacher-trajectory generation prompts used to instantiate π_T on zero-answer groups; and *(iv)* the experiment-oriented filter prompt used during dataset construction (Section 3, Experiment-Oriented Filtering) to retain candidates that genuinely involve experimental scientific inquiry. We briefly state the role of each prompt below and refer the reader to the corresponding figure for the verbatim text.

(1) Answer Judge Prompt. Figure 17 shows the prompt used by the LLM-as-Judge pipeline that produces R_{answer}. The primary judge (`gpt-4.1-mini`) emits a single token in {`correct`, `incorrect`, `unknown`}; `unknown` samples are forwarded to a secondary judge (`gemini-3-flash`), and the residual $\sim 5\%$ that remain `unknown` are scored as 0. The same pipeline is shared between training-time reward computation and benchmark-time evaluation.

(2) Analogical Reasoning Reward Prompt. Figure 18 shows the prompt used by the reward model to score post-retrieval reasoning along three independent axes – *structural identification*, *role mapping*, and *procedural transfer* – each on a binary $\{0, 1\}$ scale. The final reward is $R_{\mathrm{analogy}} = \max(s_{\mathrm{id}}, s_{\mathrm{map}}, s_{\mathrm{trans}})$, and is then subject to the same outcome-conditioning rule as R_{tool}.

(3) Teacher Trajectory Generation Prompt. Figure 19 shows the prompt that instantiates the teacher policy π_T (`gpt-5.2`) on zero-answer groups. Three design choices distinguish it from a generic agent prompt and align its output with what ARIS's imitation branch needs. *(i) Demonstration framing.* The role line frames π_T as producing high-quality demonstration trajectories rather than minimal-token correct answers, so the resulting y^* is shaped to be imitable by the student instead of optimised for terseness. *(ii) Analogical-reasoning paradigm.* An explicit paradigm sentence directs the teacher to retrieve a structurally analogous exemplar via `exemplar_recall` *first*, then identify its solution schema, map its roles, and transfer the procedure; the <THINK> contract is correspondingly extended to require an explicit naming of the structurally analogous exemplar and of how its schema / roles / procedure transfer to the target. This mirrors the three axes $(s_{\mathrm{id}}, s_{\mathrm{map}}, s_{\mathrm{trans}})$ on which the student is later graded, so each demonstrated step reinforces the exact behaviour the reward model rewards. *(iii) Partial ground-truth injection.* The reference answer is injected as a separate `ground_truth` field that is visible to the teacher only, with explicit anti-leakage constraints: it must

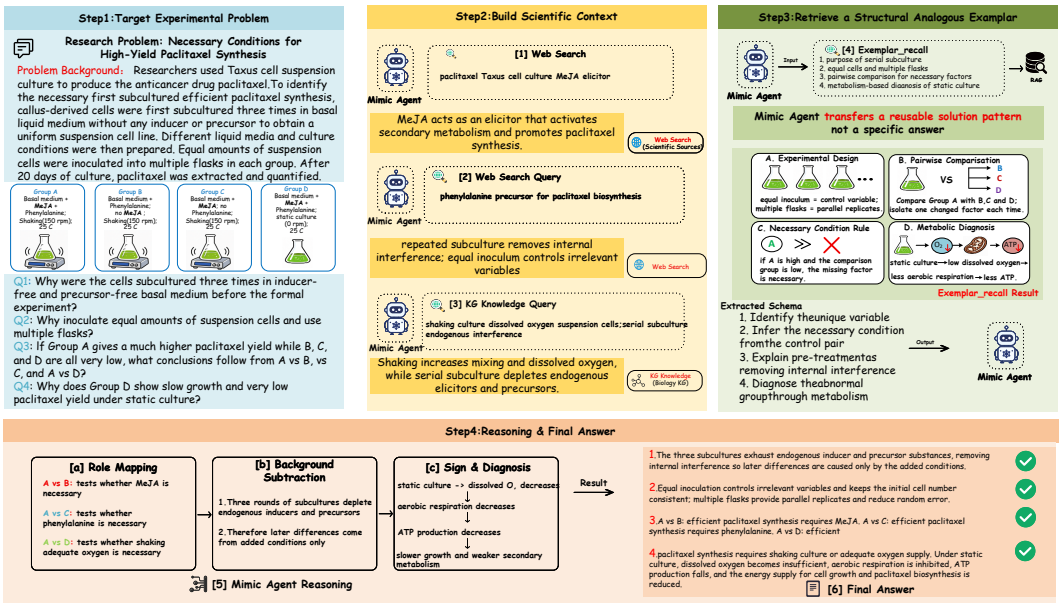


Figure 16: Case study (biology): MimicAgent solves a paclitaxel multi-factor necessity experiment in *Taxus* suspension cultures. The trajectory shows tool-augmented background building (MeJA as an elicitor; phenylalanine as a precursor; aerobic-respiration / dissolved-oxygen pathway via `kg_knowledge`), exemplar retrieval of a structurally analogous chlorophyll-synthesis condition test, schema extraction (*multi-factor necessity* + *endogenous-reserve depletion* + *single-factor knock-out*), and the final reasoning aligning each control group to one necessary factor.

keep the trajectory on track but cannot be copied into <THINK> or tool arguments and cannot be revealed before the final <ANSWER> step. This injection raises the success rate of π_T on otherwise zero-answer prompts (which are by definition the hardest, and also the most expensive to recover purely from π_T's parametric knowledge) without turning y^* into an answer-conditioned dump that the student cannot plausibly reproduce. The prompt also shares the tool set $\mathcal{T}$ and formatting constraints $\mathcal{F}$ with the student prompt, so every action in y^* remains valid under the student's tool schema, and ARIS's token-masked imitation loss can be applied directly to model-generated tokens. Empirically, these three choices together concentrate the teacher's output on schema-bearing reasoning steps that the student's imitation loss can pick up, effectively raising the per-token information density of the rare imitation updates triggered by ARIS.

(4) Experiment-Oriented Filter Prompt. Figure 20 shows the prompt used during dataset construction (Section 3) to retain candidates that genuinely involve experimental scientific inquiry. The filter (`gpt-5.2`) scores five independent binary dimensions $(d_1, \ldots, d_5)$ corresponding to (i) experimental design, (ii) analysis requirement, (iii) experimental setup, (iv) data interpretation, and (v) process reasoning, and emits a single JSON line containing the five scores together with a final `decision` of `keep` (if at least one $d_i = 1$) or `discard`. Constraining the output to a single JSON line and forbidding inferred content keeps the filter strictly grounded in what the question itself states.

E.2 Consistency Analysis

Our reward pipeline relies on external models in two places: (i) R_{answer} uses an LLM-as-Judge to determine final-answer correctness, and (ii) the three sub-scores of R_{analogy} (structural identification, role mapping, and procedural transfer) are produced by a separate reward model. We assess the consistency of these two external signals with human judgment via a small-scale annotation protocol, and Table 8 summarizes the results.

(a) Answer judge consistency. We randomly sample 200 (`prompt, model_answer`) pairs from the Chinese test split of SciExplore-Bench, and ask three independent annotators (each holding at least a bachelor's degree in a relevant scientific discipline) to label each pair as `correct/incorrect`. The majority vote serves as the human reference; the pairwise Cohen's κ averaged across the three

Answer Judge Prompt

```
You are an impartial evaluator. Judge whether a model's answer matches the
reference answer for the given multimodal scientific question, and output a
single token only.

Judging rules:
- Output exactly one of: correct / incorrect / unknown.
- Ignore case, whitespace, punctuation, and minor formatting differences (e.g.
  semicolons vs. commas, trailing units, equivalent symbol forms).
- If the model answer is empty, or you cannot decide whether it is equivalent to
  the reference, output unknown.
- Do not output anything else (no rationale, no quoting, no Markdown).

Question:
{question}

Reference answer:
{ground_truth}

Model answer:
{model_answer}

Respond now with exactly one word: correct / incorrect / unknown.
```

Figure 17: Answer judge prompt.

annotators is $\bar{\kappa}_{\text{human}} \approx 0.80$, which we report as the inter-rater baseline. On the same pairs we evaluate the two judge models actually used in the R_{answer} pipeline (shared between training-time reward computation and benchmark evaluation): `gpt-4.1-mini` (primary judge) and `gemini-3-flash` (secondary judge, invoked only when `gpt-4.1-mini` returns `unknown`). Both are compared against the human reference; the proportion of pairs on which each judge emits `unknown` is roughly 5% and 3% respectively.

(b) Analogical-reward consistency. We further sample 150 rollout trajectories that invoke `exemplar_recall`, and the same annotators independently score each trajectory along structural identification, role mapping, and procedural transfer using a binary $\{0, 1\}$ scale; the three-annotator majority vote serves as the human reference and is compared with the `gemini-3-flash` reward model's $\{0, 1\}$ outputs $s_{\text{id}}, s_{\text{map}}, s_{\text{trans}}$ on the same dimensions. The reward model attains an overall Pearson $r \approx 0.75$, with per-dimension Spearman $\rho_{\text{id}} \approx 0.75$, $\rho_{\text{map}} \approx 0.71$, and $\rho_{\text{trans}} \approx 0.80$; mean absolute error (MAE) is reported on the $[0, 1]$ scale.

Table 8: Consistency between automatic reward signals and human judgment under our annotation protocol.

Signal	**Cohen's κ / Pearson r**	**Accuracy / Spearman ρ**	**Unknown rate / MAE**
R_{answer} primary judge (`gpt-4.1-mini`)	$\kappa \approx 0.79$	accuracy $\approx 88\%$	unknown rate $\approx 5\%$
R_{answer} secondary judge (`gemini-3-flash`)	$\kappa \approx 0.82$	accuracy $\approx 90\%$	unknown rate $\approx 3\%$
R_{analogy} reward model (`gemini-3-flash`), pooled across dims	$r \approx 0.75$	$\rho \approx 0.74$	MAE ≈ 0.15
structural identification s_{id}	$r \approx 0.74$	$\rho \approx 0.75$	MAE ≈ 0.15
role mapping s_{map}	$r \approx 0.70$	$\rho \approx 0.71$	MAE ≈ 0.17
procedural transfer s_{trans}	$r \approx 0.80$	$\rho \approx 0.80$	MAE ≈ 0.12

Analogical Reasoning Reward Prompt

```
You are a strict scientific-reasoning evaluator. You will see (1) a TARGET
scientific question, (2) a set of EXEMPLARS retrieved by an agent through the
exemplar_recall tool, and (3) the agent's POST-RETRIEVAL reasoning. Decide
whether the post-retrieval reasoning shows imitation-driven transfer from the
exemplars to the target. Score three independent axes, each as a binary 0 or 1:

s_id (Structural Identification): does the reasoning clearly state the relevant
solution schema / template / canonical procedure of the exemplar?
  1 = yes, the schema / equation / procedure is explicitly stated
  0 = otherwise

s_map (Role Mapping): does the reasoning systematically align the key variables,
entities, or experimental roles between the exemplar and the target?
  1 = yes, the key variables / roles are systematically aligned
  0 = otherwise

s_trans (Procedural Transfer): does the reasoning apply the exemplar's procedure
end-to-end and yield valid downstream steps for the target?
  1 = yes, the full procedure is executed and produces valid intermediate
      results
  0 = otherwise

Rules:
- Score only on what actually appears in POST-RETRIEVAL reasoning; ignore the
  final answer's correctness.
- The three axes are independent; a high score on one does not imply high scores
  on the others.
- Output STRICT JSON with exactly the keys s_id, s_map, s_trans, each an integer
  in {0, 1}. No prose, no extra keys.

[Target Question]
{question}

[Retrieved Exemplars]
{exemplars_block}

[Post-Retrieval Reasoning]
{traces_block}

Output format (verbatim):
{"s_id": <int>, "s_map": <int>, "s_trans": <int>}
```

Figure 18: Analogical-reasoning reward prompt.

Teacher Trajectory Generation Prompt

```
You are a rigorous scientific reasoning agent that produces high-quality
demonstration trajectories by solving problems through analogy to retrieved
exemplars - please keep working until the user's question is completely resolved
before ending your turn. You may call external tools to gather verifiable
evidence and ground every reasoning step on the question, the attached image(s),
and tool-returned evidence.

You will be solving multimodal scientific inquiry problems that span
experimental design, variable control, evidence interpretation, and mechanism
inference. Prefer exemplar-guided analogical reasoning over parametric recall:
retrieve a structurally analogous companion exercise, then identify its solution
schema, map its roles to the target problem, and transfer its procedure. Do not
invoke external knowledge that you cannot ground in the question, the image(s),
or tool responses.

You have access to the following tools, defined as JSON schema:
{func_schemas}

Tool-use protocol:
- Tools are optional. If you can answer reliably without one, do so.
- exemplar_recall returns K=3 curatorially aligned companion exercises with
  reference solution traces; call it early and organize the rest of your
  reasoning around the retrieved schema.
- web_search retrieves open-domain web evidence.
- kg_knowledge queries the structured knowledge graph.
- image_crop returns a localized crop of an image.
- code_interpreter executes Python in a sandbox (numpy / scipy / sympy /
  matplotlib / pandas pre-loaded).
- A tool result is returned inside <tool_response>...</tool_response>; treat its
  content as the only verifiable evidence for that step.
- At most 6 tool-call rounds; each round invokes at most one function.
- After emitting <tool_call>, end this reply and wait for the next
  <tool_response>.

Output format (strict, must be obeyed):
- Every reply consists of exactly two adjacent segments and nothing else (no
  Markdown, no extra text):
    A) When calling a tool: <think>...</think><tool_call>{...}</tool_call>
    B) When committing to a final answer:
       <think>...</think><answer>\boxed{...}</answer>
- <think> states what is known, what is missing, which retrieved exemplar is
  structurally analogous, how its schema / roles / procedure transfer to the
  target, and why the next step is the chosen tool call or final answer.
- Once you can answer, do not call further tools.

Image resolution:
Image Width: {width}; Image Height: {height}

User question:
{question}

Reference answer (visible to you only, for trajectory guidance):
{ground_truth}
Use the reference answer to keep the trajectory on track; do not copy it
verbatim into <think> or tool arguments, and do not reveal it before the final
<answer> step.

Respond now in the format above, ending with <answer>\boxed{...}</answer>.
```

Figure 19: Teacher trajectory generation prompt. The prompt frames π_T as producing demonstration trajectories rather than minimal answers, enforces an exemplar-first analogical-reasoning workflow (retrieve $\rightarrow$ identify schema $\rightarrow$ map roles $\rightarrow$ transfer procedure) that mirrors the three reward axes $(s_{\mathrm{id}}, s_{\mathrm{map}}, s_{\mathrm{trans}})$, and injects the reference answer as private guidance for the teacher only, with explicit anti-leakage constraints. Together these choices yield expert trajectories that are simultaneously schema-bearing, valid under the student's tool schema, and directly absorbable by ARIS's token-masked imitation loss.

Experiment-Oriented Filter Prompt

```
You are a strict experimental-science filter. Given a multimodal scientific
question, decide whether it genuinely involves experimental scientific inquiry,
and output a single JSON line only.

Filtering rules:
- Score five independent dimensions, each in {0, 1}:
  - d1 (experimental design): presence of experimental design, procedures, or
    operations.
  - d2 (analysis requirement): the question requires proposing plans, predicting
    outcomes, or analyzing observations.
  - d3 (experimental setup): explicit experimental materials, conditions, or
    variable-control structures.
  - d4 (data interpretation): the question requires interpreting data and
    drawing scientific conclusions.
  - d5 (process reasoning): explicit reasoning over experimental processes.
- Decision is "keep" if at least one of d1..d5 equals 1; otherwise "discard".
- Score only on what actually appears in the question; do not infer.
- Output STRICT JSON with exactly the keys d1, d2, d3, d4, d5 (each integer in
  {0, 1}) and decision (one of "keep" / "discard"). No prose, no extra keys.

Question:
{question}

Output format (verbatim):
{"d1": <int>, "d2": <int>, "d3": <int>, "d4": <int>, "d5": <int>, "decision":
"<str>"}
```

Figure 20: Experiment-oriented filter prompt.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and the contribution list at the end of Introduction explicitly state our three contributions (SciExplore-Bench, MimicAgent, and ARIS).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The main limitations of our approach are discussed in the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Theorems and proofs are included in the Method section.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All training and evaluation details (model backbone, optimizer, learning rate, batch size, decoding configuration, tool budget, reward weights, and ARIS hyperparameters) are documented in the Implementation Details and the appendix; code and prompts are provided in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data and code are organized in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The Implementation Details paragraph in the appendix fully specify the training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following [27], we perform a single-pass evaluation on SceMQA. On SciExplore-Bench, we run each experiment five times and report the average performance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All training experiments are conducted on a single node with $8\times$NVIDIA H100 (80 GB) GPUs; per-experiment runtimes and total compute are reported in the appendix Section A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics: no scraped personal data, no high-risk generative content, and all annotation in the consistency analysis is performed internally by the authors and qualified scientific collaborators rather than crowd workers.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational research on tool-augmented scientific reasoning evaluated on educational science problems; it does not introduce a deployed system or generative artifacts with a direct path to negative applications such as disinformation, surveillance, or biased decision-making.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: All data and models used in this work are publicly available online and can be accessed free of charge.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All baseline models (InternVL2.5, Qwen2.5-VL, Qwen3-VL, GPT-5/5.2, Claude-Haiku-4.5, Gemini-2.5/3) and external evaluation datasets (SceMQA, MM-IQ, VisualPuzzles) are properly cited at point of use; their respective publications and license terms are referenced via the bibliography.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release SciExplore-Bench together with the analogical exemplar repository, training and evaluation scripts, prompt templates, and a README documenting reproduction commands as part of the anonymized supplementary material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. The internal small-scale rating used in the Consistency Analysis is performed by qualified scientific collaborators reviewing AI-generated outputs as part of the authors' research workflow, not crowdsourced labor.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Our small-scale annotation study does not constitute human subjects research and is exempt from IRB review under the relevant institutional policies; participants reviewed AI-generated content rather than personal data and were not subject to risk.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are core components of our method.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.